UNIVERSITA' DI PADOVA FACOLTA' DI INGEGNERIA

Dipartimento di Ingegneria dell'Informazione

Scuola di Dottorato di Ricerca in Ingegneria dell'Informazione
Indirizzo: Scienza e Tecnologia dell'Informazione

CICLO: XXIII

A solution to the problem of the generalization of the Italian geographical databases from large to medium scale: approach definition, process design and operators implementation

Direttore della Scuola: Ch.mo Prof. Matteo Bertocco

Supervisore: Ch.mi Proff. Massimo Rumor, Sergio Congiu, Matteo Bertocco

Dottorando: Sandro Savino

"fatti non foste a viver come bruti,
ma per seguir virtute e canoscenza"
(Dante Alighieri, Divina Commedia,
Inferno canto XXVI, 116-120)

To my family and my friends.
Sandro Savino, 2011

Abstract

During many years in which the generalization of cartographic data has been studied many developments have been achieved. As some national mapping agencies in Europe and in the world are beginning to introduce automated processes in their production lines, the original dream of a completely automated system that could perform generalization is getting closer, even though it has not been reached yet.

The aim of this dissertation is to investigate whether it is possible to design and implement a working generalization process for the Italian large-medium scale geographical databases.

In this thesis we argue that the models, the approaches and the algorithms developed so far provide a robust and sound base to the problem of automated cartographic generalization, but that to build an effective generalization process it is necessary to deal with all the small details deriving from the actual implementation of the process on defined scales and data models of input and output.

We speculate that our goal can be reached by capitalizing the research results achieved so far and customizing the process on the data models and scales treated.

This is the approach at the basis of this research work: the design of the cartographic generalization process and the algorithms implemented, either developed from scratch or deriving from previous works, have all been customized to solve a well defined problem: i.e. they expect input data that comply to a consistent data model and are tailored to obtain the results at defined scale and data model.

This thesis explains how this approach has been brought into practice in the frame of the CARGEN project that aims at the development of a complete cartographic process to generalize the Italian medium scale geographical databases at 1:25000 and 1:50000 scales from the official Italian large scale geographical database at 1:5000 scale. This thesis will focus on the generalization to the 1:25000 scale, describing the approach that has been adopted, the overall process that has been designed and will provide details on the most important operators implemented for the generalization at such scale.

Sommario

L'argomento di questa tesi di dottorato è la generalizzazione cartografica automatica, applicata ai database geografici italiani alla media e alta scala.

Il lavoro di ricerca sulla generalizzazione cartografica automatica, frutto di oltre 40 anni di studio a livello internazionale, ha portato a numerosi ed importanti sviluppi nel campo, recentemente concretizzatisi nella scelta di alcuni enti cartografici nazionali di adottare sistemi di generalizzazione automatica nei propri processi produttivi. Nonostante i continui progressi e i positivi risultati della ricerca, però, il traguardo di un processo di generalizzazione completamente automatico non è ancora stato raggiunto.

L'obiettivo di questa tesi è di indagare la possibilità di implementare un processo automatico di generalizzazione cartografica per i database geografici italiani alla media e alta scala.

La tesi si basa sull'ipotesi che i modelli, gli approcci e gli algoritmi proposti finora costituiscano una base solida da cui muovere per affrontare il problema della generalizzazione cartografica, ma che per sviluppare un processo di generalizzazione completo sia necessario sviluppare tecniche adatte specificamente ai requisiti, alle specifiche e alle particolarità dei dati da generalizzare. La nostra ipotesi è quindi che il processo di generalizzazione possa essere realizzato a partire dai risultati della ricerca adattando il processo alla scala e ai modelli dati specifici del nostro problema.

Questo è l'approccio alla base di quanto sarà esposto in questa tesi: il processo di generalizzazione e gli algoritmi sviluppati, o modificati da lavori esistenti, sono stati tutti progettati per risolvere una specifica parte del nostro processo di generalizzazione.

Il lavoro di ricerca presentato in questa tesi è stato sviluppato all'interno del progetto CARGEN, un progetto di ricerca tra l'Università di Padova e la Regione Veneto, con la collaborazione dell'IGMI, per lo sviluppo di una procedura automatica di generalizzazione del database DB25 IGMI in scala 1:25000 a partire dal database regionale GeoDBR in scala 1:5000.

Il lavoro di tesi affronta tutti i temi relativi al processo di generalizzazione, partendo dalla generalizzazione del modello fino alla descrizione degli algoritmi di generalizzazione delle geometrie.

Thanks to

Three years of PhD may seem a very short time, especially when they do not stand anymore in front of you but they are behind your shoulders. Indeed, they are a very long period of time when it comes to thank all the people met along the way. Maybe I was a lucky one, but the quantity of great people I met and I had the chance and pleasure to work, talk, reason, discuss, program and -why not?- have a beer or two with and that now I would like to thank, simply outnumbers my chances to make this thing right, without forgetting anyone.

As it is, I will then simply make a brief list of people, places and events, and whoever feels that his name should be in the list, please do not bear me any ill will but just take a pen and write it here below.

I would like to thank all the people that work and worked at the CARGEN project, starting with professor Massimo Rumor and Sergio Congiu, Damiano Salazzari and Silvia Dalla Costa, the people at IGMI and Regione Veneto Gennaro, Cinzia, Geremia, Claudio, Maurizio, Antonio, Pierpaolo, and then all the dozens of students, Italians and foreigners, who contributed at the research like Elia, Rossella, Matteo, Marco, Damiano, Igor, Andrea (and many many others). I also would like to thank other people at DEI, the Department of Information Engineering: both those who worked in our lab, making the life there an enjoyable experience, and also other PhD students, like Riccardo, Gianmaria, Emanuele.

I would like also to thank all the people of the TU-Delft who made my six months there a terrific time: Martijn, Hugo, Wilko, Edward, Elfride, Peter and all the people of the GIS-Technology section; also I owe much to my flatmates and friends Eva, Tine, Felicity, Eelco, Jesus and so on …thanks to all of them there in the Netherlands I could grow and learn about both research and real life.

Then I would also like to thank all the people met at the conferences that I enjoyed so much to attend to, especially those at ICA and UDMS.

Last but not least, I want to thank my family and friends, who know that I owe them a lot.

Contents

Chapter 1

Introduction

This dissertation focuses on the problem of cartographic generalization.

This work is part of the CARGEN project, a research project of the Department of Information Engineering of the University of Padua, aimed at the design and implementation of an automated generalization process to derive Italian geographical databases at the scale 1:25000 and 1:50000 from the scale 1:5000.

The automation of cartographic generalization is a very complex and broad research topic; as a consequence, it can not be fully addressed in the three years course of a single PhD thesis.

For this reason, of all the work done in the CARGEN project, this thesis will narrow its focus only on the generalization to the 1:25000 scale, and in particular will describe the approach that we adopted in our research and the overall process that has been designed; moreover, it will provide details on the most important operators implemented in the project for the generalization at such scale.

Being part of a broader research effort, in some cases the material presented in this thesis is based on works that have been developed together with other authors: in the case explicit credits to them will be given along the text.

As the CARGEN project is still under development at the present day, the results presented in this thesis can not be considered conclusive. The approach adopted, the process designed and the operators implemented are all subject to further development and improvements; although we consider what is presented in this thesis to represent more than just a partial solution, it can not be considered a complete solution to the complex problem of cartographic generalization and leaves space to further improvements.

The scientific contributions of this work to the research field on cartographic generalization can be summarized as:

the development of some novel generalization solutions (e.g. road junction simplification, ditches typification);

the design and partial development of a complete process to generalize Italian large-medium scale data.

Table 2 in chapter 6 summarizes the algorithms developed, pointing out the original solutions; a more broad discussion of the contributions brought by this thesis can be found in chapter 8.

Next, in this brief introductory chapter, a definition of cartographic generalization is given, along with an overview of its benefits. Following it the synopsis of this thesis closes this chapter.

1.1 Cartographic generalization

During the International Cartographic Association conference of 1973, cartographic generalization has been defined as: "the selection and simplified representation of detail appropriate to scale and/or the purpose of a map".

In other words, cartographic generalization is the process used to produce a new map using the data of an existing cartography: usually the process involves an input at a larger scale, thus containing more detail, to derive an output at a smaller scale.

Cartographic generalization has two key benefits:

the first is that it allows to use existing data to produce a cartography, thus reducing the costs (in general terms of resources) of map production but also allowing the creation of maps that are "synchronized", i.e. that represent the same space (territory) at the same moment[1]

the second is that it allows to represent information in a more compact way, this being useful either to represent more data in the same space (what is done when representing the same territory at two different scales) or to represent the same data in a smaller space (this is useful to produce maps that fit smaller media, e.g. a pc monitor or a mobile, see [Gimodig, 2001])

Cartographic generalization has been done extensively by hand in the past, thus reducing its beneficial impact; only the automation of this process would allow to exploit completely its benefits. As map making, generalization is a very complex task and many years of research proved that its automation is a task at least as complex, if not more; despite this, the research is continuously progressing and leading to more and more concrete results. The challenging aspects to overcome to achieve automated cartographic generalization and the important benefits that this could bring make the research in this field both very interesting and exciting.

[1] Usually maps at different scales have different update cycles (larger scales being updated faster than smaller scales) and require a different amount of time for their creation. With traditional map making techniques the effect of this is that usually two different maps of the same area are created in two distinct time frames and as the territory represented could have changed significantly between the creation of the first and the second map, there can be some inconsistencies between the two. With generalization this problem is overcome as every map is derived from source data collected in a single time frame.

These same motivations are at the base of this research work and of the CARGEN project.

1.2 Thesis Overview

This thesis contains eight chapters with the following content:

Chapter 2 is about cartographic generalization. This topic is covered with a brief overview of the early stages of research, leading to the first models and algorithms. Then the present state of the research in generalization is presented; the section focuses on the most important works in the field, highlighting the approaches developed, the software available and the real implementations.

Chapter 3 sets the background of this research work. The present situation of cartography in Italy is illustrated and the CARGEN project is introduced. The chapter then explains the approach adopted and the first design choices taken at the beginning of the project. The developments deriving from these choices constitute the main body of this research work and are given in the following chapters.

Chapters 4, 5 and 6 explain in details the generalization process.
Chapter 4 is about the overall generalization process: here the choices made to design the generalization process are further illustrated and explained. Some relevant peculiarities of the project are highlighted and then the overall process is described, modeled as an ordered sequence of steps.

Chapter 5 illustrates the model generalization process. An explanation of the purpose of this process is given, with a brief description of the general issues related to it. The models of the two geographical databases for large and medium scale involved in the process, the DBT in 1:5000 scale produced by the Regions and the DB25 in 1:25000 scale produced by the IGMI (Istituto Geografico Militare Italiano) are introduced, highlighting the main aspects and differences between them. Then the chapter focuses on how the process has been dealt with in the CARGEN project, dividing it in two tasks: matching and rule building. Both tasks are explained, describing the major issues that had to be solved. The chapter is closed by an explanation of the functions that had to be developed to perform the model generalization and some examples of their use are given.

Chapter 6 describes the generalization algorithms developed. The explanation is organized in sections, each of them describing the algorithms that have been developed to solve a specific generalization problem or the generalization of a specific feature class. For each topic the problems to solve are explained in details, together with eventual related work, the approach used and the implementation. Design choices and parameters are discussed for each algorithm, while the evaluation of the results is left to chapter 7. Despite the fact that all the algorithms have been developed to pursue the generalization of data from and to a specific data model and scale, the implementation of some of them can be adapted also to more general contexts; at the end of the chapter the algorithms are then grouped as operators on the base of the transformation they perform (e.g. typification, simplification, selection, ...) and it is suggested how they can be used on different data model or scale.

Chapter 7 shows he results obtained by the developed solution. This chapter briefly explores the topic of the assessment of cartographic generalization and describes how errors are handled in the process. Some indications of the performances of the process are given; following the results of the process are presented and discussed, highlighting the limitations and the advantages both of the single algorithms developed and of the whole process.

Chapter 8 contains the conclusions of this research work. The research statement and the results obtained are discussed. The flaws in the process and its limitations are used to trace the direction of the future developments.

The list of publications and the references used closes the thesis.

Chapter 2

Research on Cartographic Generalization

This chapter will give a brief overview of the early stages of research on cartographic generalization, leading to the first models and algorithms. Then the present state of the research in generalization will be presented; the chapter will outline the most important works in the field, highlighting the approaches developed, the software available and the real implementations.

This chapter provides a general overview of the topic of cartographic generalization: further information can be found in chapter 6, where for each step of the generalization process the most relevant related works are discussed.

2.1 Generalization

The creation of a map is a very complex task, comprising many different activities; among these, making a map requires to abstract the reality, extract those aspect of it that are most relevant to the purpose of the desired map and represent them in a symbolic form that ideally conveys the same information of the original phenomena. This process can be defined as generalization, i.e. "the selection and simplified representation of detail appropriate to scale and/or the purpose of a map" according to the definition given by the International Cartographic Association in 1973.

Depending whether the map is created from scratch, using reality as the source data to be represented, or using an already existing map, map making can be distinct in map compilation (the former) or map derivation (the latter); either way, map making is closely related to the process of generalization. In this thesis we will focus on the process of map derivation and the term (cartographic) generalization should be referred to this context; furthermore the map derivation should be always intended to take place from a source scale to a smaller target scale.

The definition of cartographic generalization given above clearly refers to two activities: selection and representation.

In fact, given a source map, its generalization to a target map with a different scale or purpose requires to choose which objects of the source map should be present in the target map -or more in general what information of the source should be present in the target- and also to decide how to represent the selected information in the target map. These two operations are also called, respectively, model-oriented and graphic-oriented generalization[2].

The process of cartographic generalization is very complex and has been proven to be extremely difficult to automate, as it requires skills that do not belong to computers. The history of the research on this field will be outlined next.

2.2 A brief history

The benefits of cartographic generalization, first of all the reduction of the costs to produce a map, pushed the research on its automation since the introduction of computers in cartography.

The beginning of the research can be set around the 1960 and its past can be divided mainly in periods [Kilpelainen and Sarjakosky, 1995; Meng, 1997; Sarjakoski, 2007] each of them being characterized by a main direction of research and a different way to approach the problem of automated cartographic generalization.

According to Meng "research activities have experienced a major cycle of upswing (e.g. 1965-1980), euphoria and suspicion (e.g. 1980-1990), and stagnation (e.g. 1990-1995) followed by possibly a new upswing (since 1995)" [Meng, 1997, p.13]

The first period, from 1960 to the late 1970, saw the birth of the first models to conceptualize the process of generalization. The model of Ratajski dates to this period [Ratajski, 1967]. According to this model, generalization consists of quantitative generalization, i.e. a gradual reduction of map content, and qualitative generalization, i.e. a transformation of the representation of map content. Generalization can be performed reducing gradually the map content (quantitative generalization) until the capacity of the map is reached. At this point, called generalization point, the content can not be further reduced without losing important information: to generalize any further it is necessary to operate a transformation of the representation (qualitative generalization); this yields to an increase in map capacity, allowing to iterate again the process. This process is shown in Figure 1: on the left the triangle depicts the map capacity, on the right it is possible to see t

---

[2] To avoid the confusion brought by the common use of the word cartographic generalization to indicate both the process of map derivation and a part of the same process (see [Gruenrich, 1985]), in this thesis we will use the terms model-oriented and graphic-oriented generalization [Weibel, 1995] for the terms model generalization and cartographic generalization; the word cartographic generalization, as map generalization and, more in general, generalization, will be used to refer to the complete process of map derivation.

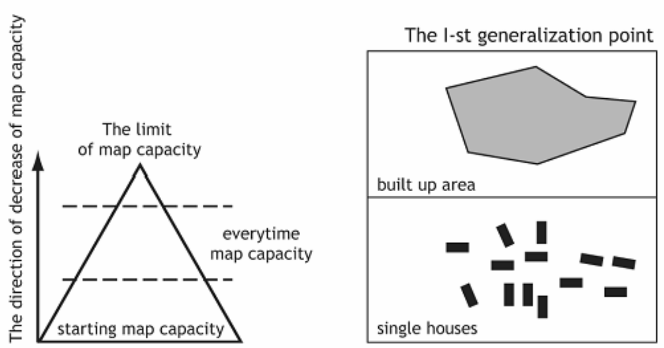


Figure 1: Ratajski's model

The research in the first period also focused on the development of algorithms for selection and simplification.

The Radical Law of Topfer and Pillewizer [Topfer and Pillewizer, 1966] was one of the outcomes of this research. Their work related the number of symbols on a map to the map scale, thus providing a parameter to tune selection algorithms, although their method did not contain any indication on how the selection should be performed.

Another outcome of this research was the development of one of the most ubiquitous line simplification algorithm, the Douglas-Peucker algorithm, dated 1973 [Douglas and Peucker, 1973] and still being one of the most used simplification algorithms.

Around 1980, the attention of research was drawn most on modeling cartographic generalization; following the advances in database technology, the distinction between model and cartographic generalization was conceived.

One of the early works where this distinction is present is that of Gruenrich [Gruenrich, 1985]. According to his model, reality is transformed into a primary DLM, Digital Landscape Model, through the operation of object-generalization. From this first model it is possible to derive many other secondary DLM, for instance each to serve a different cartographic purpose or retaining a different level of detail, from fine to coarse. Each DLM actually stores the information that suits its purpose and scale and can be used for analysis, but it is not ready to be represented as a map; in order to do so it is necessary to transform it into a DCM, Digital Cartographic Model, through the operation of cartographic generalization. In Figure 2 it is possible to see how the different generalization operations result in different products; of these, only the DCM is suitable to be printed as a map, while the DLMs can be used to perform analysis with a GIS.

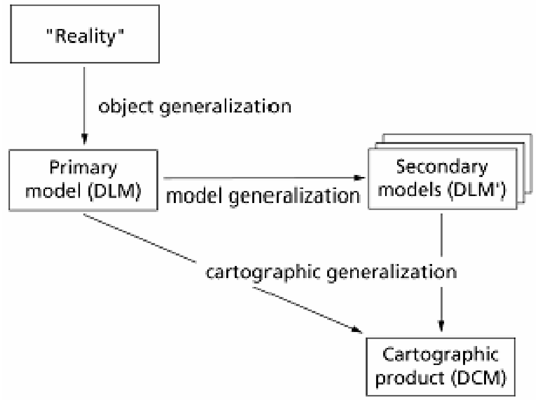
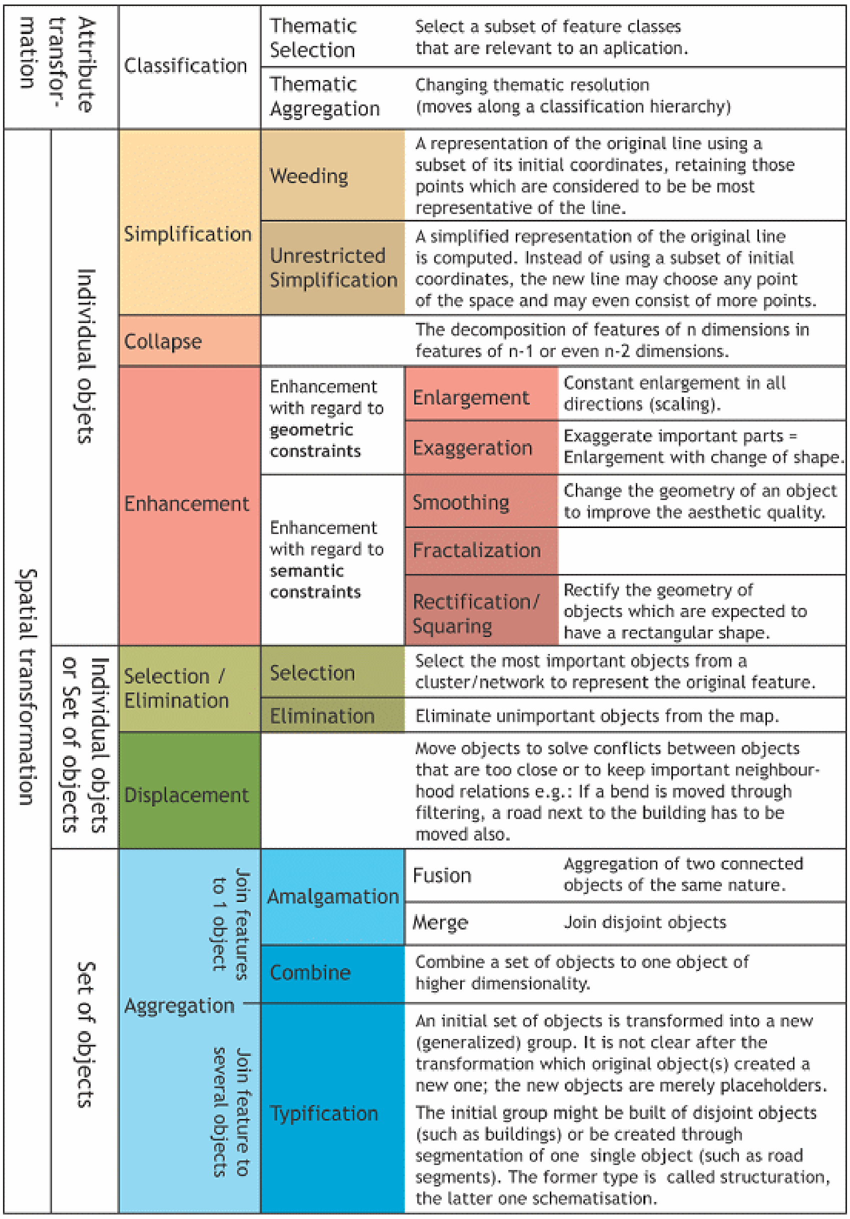


Figure 2: Gruenrich's model

In the model of Brassel and Weibel [Brassel and Weibel, 1988] the process of generalization comprised 5 steps: structure recognition, process recognition, process modeling, process execution and data display. In their work they also differ statistic generalization (later on renamed model generalization) and cartographic generalization.

McMaster and Shea analyzed the process of generalization from three separate points of view: why to generalize, when to generalize and how [McMaster and Shea, 1988]. Modeling how to generalize lead to the definition of twelve different generalization operators: simplification, smoothing, aggregation, amalgamation, merging, collapse, refinement, typification, exaggeration, enhancement, displacement and classification [Shea and McMaster, 1989]. Each operator defines a transformation either on the spatial or the semantic attributes of an object and may be implemented by one or different algorithms [Weibel and Dutton, 1999]. In a later work [McMaster and Shea, 1992], the authors modeled also when to generalize in: condition, measures and controls.

In the late 1990 a new idea allowed to model generalization as a holistic process. Generalization was modeled on constraints, i.e. particular characteristics that the generalized data should possess. Different type of constraints were identified: position, topology, shape, functional, structural and legibility [Ruas and Plazanet, 1997]. The generalization process should then try to find a generalized solution that satisfies most of these constraints. Following the model of McMaster and Shea, condition, measures and controls were used to assess the value of a constraint to check whether it was violated, and the operators were used to transform the data affecting these values in order to obtain a better generalization. The AGENT project [AGENT, 2000] is one of the most relevant examples of this approach and also one of the most valuable, as it brought to the formalization and implementation of many constraints and generalization operators (see Figure 3).



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 objects 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.
		Aggregation – 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 3: the operators in the Agent project

With the beginning of 2000, the interest in modeling the process of cartographic generalization seemed to diminish, with the attention being drawn by the research and implementation of better and more clever operators. The knowledge gathered in many years of research and the evolution of the approaches led to the development of the first actual systems for generalization.

In what we can call modern era of cartographic generalization, generalization software and systems are actually in use among some national mapping agencies (henceforth NMAs) to ease the burden of the creation of maps by partially automating the process. While no out-of-the-box solution has been created yet [Stoter, 2010], the number and range of available techniques are very high and, aside from further improving the results, the new challenge seems to be how to orchestrate all that has been done into a complete organic solution (e.g. see [Renard et al., 2010]).

2.3 Approaches to generalization

Throughout the years of research on cartographic generalization the attempts to automate this process led to the development of computer tools approaching the problem in different ways.

2.3.1 Batch

At the beginning of the research, from 1960 to late 1970, only single tools were developed, with the aim of solving some simple problems (e.g. line simplification) as an aid to the cartographers. The first generalization systems were developed as batch processes, a predefined sequence of operations iteratively run one after the other; the system did not allow to interact with the process once started, and it had to be completely repeated to change some parameters or edit the sequence of operations.

2.3.2 Condition-Action

Around 1980 the increased knowledge about generalization and the emphasis on expert systems led to the development of systems using a condition-action approach. These systems rely on a list of rules, stored in a rule base as in expert systems; each rule comprises one or more actions subject to a condition that is evaluated on the base of structural information previously gathered analyzing the cartographic data. This "structural knowledge", through the conditions, triggers different generalization actions, in a process that is more dynamic and flexible than a batch process.

2.3.3 Amplified Intelligence

Around 1990, the difficulties connected to the set up and use of expert or rule based systems (e.g. the problem of collecting and formalizing the knowledge, also known as the "knowledge acquisition bottleneck" [Weibel et al., 1995]), made many researcher to turn their attention from the elusive idea of a completely automated generalization to interactive systems: the new approach, somehow more pragmatic, was to rely on computers only for those generalization tasks that they could perform well, resorting to the human skills for the remaining ones. These interactive systems, though, are not simple editing tools comprising just a set of generalization algorithms, but are able to help and assist the user during the interactive generalization, thus augmenting his capabilities, resulting in what was called amplified intelligence [Weibel, 1991].

Unfortunately these systems proved to be not so effective in reducing the time and the resources needed in the generalization process [Ruas, 2001].

2.3.4 Constraint-based

Around the mid 1990 a new approach was starting to be evaluated: the constraint based approach. In a constraint based system the focus is not on how to perform the generalization, but on what the generalization should achieve. Constraints are usually related to cartometric measures (e.g. the minimum distance between two objects, the minimum size of an area, the minimum length of a line, ...) but also to other characteristics (e.g. the "roughness" of an object); the violation of a constraint does not trigger directly an action, as opposed to condition-action systems; instead the constraints are considered all together and the generalization is driven by a synthesis of conditions [Ruas and Plazanet, 1997]. Basically three different techniques are used to cope with all the constraints at the same time: agents, combinatorial optimization and continuous optimization. All these three techniques attempt to produce a result in which most of the constraints are satisfied; two main steps are involved in the process: first every constraint is weighted by its importance and the state of the system is evaluated by assessing which and how many constraints are satisfied, then the system performs some operations that affects those values violating the constraints until a better state is found.

Of these three approaches, the agent approach is the most versatile, as it has the potential to model all the set of operators and can ideally be adapted to handle any kind of constraint.

To date agent-based systems are the most evolved approach in generalization. Agents proved to be a very successful achievement in generalization: they are versatile, as they can be extended to solve different problems, they are flexible, as they can be thought to use different strategies to solve the same problem, they are autonomous but also interactive, as they are able to take decisions on their own but also to communicate with other agents; in general their ability to cope with many constraints make them particularly fit to handle the holistic nature of cartographic generalization. As a result they are at the core of many of the generalization solutions that NMAs are using to actually produce maps (see below).

2.4 Some present solutions

As explained previously, the constant improvement of the generalization techniques during the years finally led to the development of software and systems that are currently being used by some NMAs in their production lines.

Among the software in use, some are tailored specifically for generalization, while other provide toolboxes of generalization algorithms: among the former there are CPT (Change, Push, Tipify) developed by the University of Hannover, Radius Clarity by 1Spatial, Axpand by Axes Systems; among the latter there are ArcGIS by ESRI, Lamps2 1Spatial, DynaGEN by Intergraph.

In 2007 an European project was started to assess the “State-of-the-Art of Automated Generalisation in Commercial Software” [Stoter, 2007]; the project, ended in 2010, tested the available generalization software (CPT, Clarity, Axpand, ArcGIS) and extensively evaluated their performances. The result of the tests revealed that none of them actually provides a complete out-of-the-box solution: the softwares in some cases do not perform well or are lacking some functionalities; the result showed also that it is definitely required to customize the algorithms on the proper specifications and data models [Stoter, 2010].

Nevertheless generalization software’s are actually used by some NMAs: to overcome their limitations, each NMA developed its own generalization workflow, using some custom software and resorting on human intervention to solve the most difficult cases and to supervise and correct the automated process.

A brief outline of the systems developed by some of these NMAs will be given below, as to witness how all the efforts done in the many years of research on cartographic generalization are now bringing some tangible results. As almost every NMA is doing research in this field, the list will present only some of the experiences in this field, focusing on those actually employing automated generalization software in their production lines and highlighting the range of systems adopted for this purpose, enforcing the idea that at the moment an unique best solution to the problem does not exist.

A deep analysis of the systems developed goes beyond the scope of this thesis: for further information on the topic the interested reader is invited to consult [Stoter, 2005],[Stoter, 2010].

The ICC (Institut Cartogràfic de Catalunya) is using automated cartographic generalization since many years. They use a geographical database in 1:5000 scale to derive both the 1:10000 scale map [Baella and Pla, 1999] and the 1:25000 database [Baella and Pla, 2003]. The generalization process relies on the software CPT, on software developed by ICC and also on manual intervention; the process is both automatic and interactive, with an important percentage of the development resources invested in the implementation of interactive tools. The results of the automatic generalization process are very good, fulfilling the user requirements and bringing a three-fold increase in the productivity over traditional map-compilation. Following these results, ICC is currently reconsidering its production workflow: as it is too expensive to update all the datasets while maintaining the coherence between them in the new workflow, the 1:50000 database will be eliminated and the smaller scale maps should be generalized from the 1:25000 database [Baella and Pla, 2005].

The French IGN (Institut Géographique National - French National Mapping Agency) has a long history of research in the field of cartographic generalization, run by the COGIT laboratory. To date, as the results of the project Carto2001, started in 1999 and completed in 2005 [Lecordix et al., 2005], the IGN is using an automated generalization process to produce the Topo100 map in scale 1:100000 from the reference database BDCarto in scale 1:50000 [Jahard et al., 2003]; another research project, called New Base Map Project and started in 2004, lead to the development of a system to generalize 1:25000 and 1:50000 maps from the reference database BDTopo at 1:5000 scale [Braun et al., 2007]. Both process are developed on top of the 1Spatial Clarity and Lamps2 environment, thus adopting an AGENT based approach [Lecordix et al., 2006]; many custom algorithms have also been developed by the COGIT laboratory to improve the performance of the system on special cases (e.g. see [Gaffuri, 2007]) and one of the fields on which the very active research of IGN is focusing on is how to create a whole generalization process combining the many solutions developed (e.g. see [Touya, 2008]).

The Ordanance Survey is focussing his research on the derivation from their OS MasterMap database, storing topographic data captured at 1:1250 scale in urban areas, 1:2500 scale in rural areas and 1:10000 scale in mountain and moorland areas. One direction of research is to derive the Landranger serie at 1:50000 scale [Revell et al., 2006]: the approach adopted is based on the software Clarity, but also uses other techniques and self-developed code [Revell et al., 2005]. Recently the Ordanance Survey released a prototype of the VectorMap District serie at the 1:25000 scale that was almost completely generalized automatically [Revell, 2010].

The Danish KMS (Kort & Matrikelstyrelsen - Danish National Survey and Cadastre) employs an automatic generalization process to derive 1:50000 scale maps from the national digital topographic base map Top10DK at 1:10000 scale [West-Nielsen and Meyer, 2007]. The generalization process relies on 1Spatial software for the generalization of data and Label-EZ from MapText [Label-EZ, 2005] for label-placing. The generalization process uses both a sequential method, where each theme or layer is generalized separately in a sequence and a context driven approach, where the generalization of the objects is influenced by their contexts. The process comprises more than one hundred methods, most of them developed by KMS to customize the process on their specific needs.

The Turkish HKG (Harita Genel Komutanligi - General Command of Mapping) set up a generalization process to derive maps at the scale 1:50000 and 1:100000 from the scale 1:25000 [Simav et al., 2010], as the result of a research project called KARTOGEN and started in 2002. The process is developed on ESRI ArcGIS software and is based on different approaches: batch processing, condition-action modeling and human intervention are all used in the generalization process. The latest research is aimed at the development and integration in the process of constraint-based techniques: recent tests to evaluate the performances of this technique in the task of label-placing proved to produce very good results.

2.5 Conclusions

In this chapter a description of cartographic generalization was given, along with a brief history of the research and the approaches developed in this field. It was shown how the process was divided in model generalization and cartographic generalization and how the transformations applied to the data can be modeled as operators.

The chapter outlined the different approaches developed by the researchers along the years; at present the constraint based approach integrated with agent systems seems the most advanced and promising one, as with its ability to handle multiple constraints at a time it is suitable to model the holistic nature of generalization.

Nevertheless the chapter highlighted how no out-of-the-box solution exists yet to the problem of automated cartographic generalization: the systems currently implemented in production workflows use a wide range of different solutions, showing that a "best" solution has not been found yet; furthermore they demonstrate that the generalization process needs to be customized on the specifications and the data models. Despite the continuous improvements brought by research, human intervention is still required in most of the systems described to correct and supervise the process, showing that, in general, further research is necessary to achieve a better automation of the process.

Chapter 3

Background, approach and design choices

This chapter will complete the background behind this research work. Whereas the first chapter introduced briefly the topic and the objective of this thesis and the second gave an overview on the present state of research in the field of cartographic generalization, this chapter will present the context of this work, with a brief introduction on the present situation of cartography in Italy and of the CARGEN project. With the context completely set, it will be possible to illustrate the first steps done: the definition of the approach and the initial design choices.

3.1 Cartography in Italy

Historically the first Italian national mapping agency was the Istituto Geografico Militare Italiano (henceforth IGMI), the cartographic branch of the Army. Born in 1872, the IGMI started its activity with the compilation of the 1:100000 scale “Nuova Carta Topografica d'Italia”, followed by the production of the Serie25V, covering the whole National territory at 1:25000 scale and, later on, by the 1:50000 scale Serie50.

Accordingly to a law of 1960, the production of cartography in Italy was assigned to two bodies: the IGMI for the medium to small scales (1:25000 and smaller), and the cadastre for the large scales (1:10000 and larger). Things changed when a law of 1977 allowed also the 20 Italian Regions to produce maps on their own. This led to the creation of Regional maps at 1:5000 scale (1:10000 for less populated areas), called “Carta Tecnica Regionale” (henceforth CTR); as a central authority to govern the production of these maps was missing, every region actually created its own map with little or no standard definitions among different CTRs.

All the maps produced at that time were paper maps drawn by hand with analogue techniques. Things changed with the introduction of computers: first the CTR were all scanned into their digital counterpart “Carta Tecnica Regionale Numerica” (henceforth CTRN), later the whole production lines slowly migrated to the use of computer and digital instruments.

In 2000 IGMI started the production of a new line of maps, the Serie25DB: among the novelties of this line was the explicit use of a geographical database to store the information, with the formal definition of a data model and database schema.

Following it, the concepts of geographical databases reached also the regional mapping agencies; although the construction of geographical databases required a further evolution of the map making process, their benefits were quite evident: the use of databases allows to overcome the limits of the CTRN as the traditional division into sheets, allows to set rules on the consistency of data and allows to define topological rules among the feature classes, leading to a higher quality final product.

At the same time, the need for a standard data model shared among all the Italian regions became evident; this led to the creation of a working group comprising both the Regions and IGMI to define a national data model. The most recent embodiment of their work is the document “Catalogo dei dati territoriali - Specifiche di contenuto per i DB Geotopografici (versione 1.0)” of February 2010 [Cnipa, 2010]. This document defines a National data model for the large scales (1:5000, 1:10000) maps and sets the minimum requirements that every regional cartography should satisfy, listing a set of “core” feature classes and attributes to be implemented. Once finally approved and adopted by the Regions, this data model will set the basis for an easy sharing of geographical data among the different regions of Italy. Moreover, as the whole national territory will be described using the same data model, this will give the opportunity to design an unique generalization process that could be applied to the data produced by any Region, making possible to generalize maps from large scale and frequently updated data covering the whole territory of Italy.

Exactly in the midst of this evolution the CARGEN project was born.

3.2 The CARGEN project

This work has been developed within a research project called CARGEN. The CARGEN project was born in 2006 as a cooperation between the Department of Information Engineering of the University of Padua and the Regione Veneto (the local government of the region where Padua is), with the collaboration of the IGMI.

CARGEN means CARtographic GENeralization, and the project original objective was the design, development and test of an automated process for the cartographic generalization of the IGMI geographical database DB25 in 1:25000 scale from the regional geographical database GeoDBR in 1:5000 scale. Due to the good results achieved, in 2009 the project was extended to cover also the generalization at a smaller scale, the 1:50000.

The far reach of the objective of the project would be to modernize the map making process in Italy: with a cartographic generalization process set up, it would be possible to increase the speed of the creation of the medium scale national cartography deriving it from the regional ones; moreover, as the cartography produced by the local administrations is updated faster than the medium scale national one, the latter would enjoy a faster update cycle; finally it would be possible to propagate the updates on the large scale maps to the medium scale ones easily, thus keeping these various scales all synchronized to each other (e.g. see [Kilpelainen and Sarjakosky, 1995], Lecordix and Lemarié, 2007]).

As a starting point, the project could rely on:

the data model and specifications of the GeoDBR in 1:5000 scale
the specifications on the IGMI DB25 geographical database and maps
a sample dataset in 1:5000 scale comprising a 326000 acres territory belonging to the "Parco delle Dolomiti Bellunesi"

The specifications in particular contained some geometrical constraints on the features of the DB25 and indicated some general rules on how to derive them from larger scale maps: although these rules were a bit "loose" and relied very much on human interpretation, they were useful as they highlighted the most important transformation to apply during the process. The project could also enjoy the guidance and expertise of the cartographers both of IGMI and Regione Veneto and, last but not least, could rely on the whole body of research done in the field of cartographic generalization.

A deep analysis of all of this led to the definition of our approach and to the initial design choices that are explained in the remaining of this chapter.

3.3 Approach and design choices

When the CARGEN project started its objective was quite clear: to develop an automated process to generalize the 1:5000 regional database to the 1:25000 scale (later also to the 1:50000). Aiming at a working solution, we decided since the beginning to adopt a very pragmatic approach: the interest was not in setting a new theoretical approach to generalization but rather to implement a process that could produce some sound results. In this perspective, to develop new models or new strategies was not seen as a main objective of the research, but only a possible way to reach the goal.

To define how to reach the goal, the study of the documents and of the state of the research was the next obvious step. From the analysis of the past research works, some considerations came out clear:

in the many years of research a big deal of work has been done and lots of interesting results have been produced,
albeit some generalization processes have been implemented and are being used in production workflows, there is not any out-of-the-box solution yet.

These considerations suggested that while the understanding of the generalization process is quite deep and the tools available are quite effective, what is needed in order to set up a working generalization process is to organize, orchestrate all the knowledge and all the generalization tools with the perspective of a customization of the process on our input and output data.

We decided then to purse our objective taking advantage of the results obtained so far, modifying the existing solutions according to our specific input and output scales and models and developing new ones if needed. As a consequence of this, the process developed in the CARGEN project and partially illustrated in this thesis is tailored on our specific input and output; although some parts of it can probably be applied in other contexts, it should not be seen as a complete solution to the general problem of cartographic generalization.

About the actual development, as customization was deemed to be a key aspect in the solution, we decided to not use any vendor software, but to develop our own, implementing all the algorithms, both new and existing, by ourselves. This enabled us to insert in the code the customization that we needed for our purposes, and freed us from external software providers.

We decided that all the algorithms were to be developed using the same programming language: this is fundamental in big software projects as it allows the re-usability of code, the growth and improvement of a shared knowledge among the programmers, to set standard procedures for programming, debugging and testing and to merge seamlessly the code developed by different programmers. The code should rely on a base of shared libraries and common functions and be organized in a set of modules, each solving a particular generalization problem: this choice allows the development of a flexible solution instead of an unique big monolithic code difficult to extend and improve.

The choice of the language fell on Java [Gosling and McGilton, 1996]: Java is a modern language, object-oriented, is quite widespread in the community of people working on generalization, can rely on great libraries (e.g. the Java Topology Suite [JTS, 2002], GeoTools [GeoTools, 2002]), can be used to develop plug-ins both for open source and vendor GIS software's (e.g. OpenJump [OpenJump, 2004], ArcGIS [ESRI, 2004]) and is supported by the majority of the spatial DBMS, as Oracle [Oracle, 2005] and PostGIS [PostGIS, 2002].

The solution developed is then completely an ad-hoc solution, carefully customized for our input and output scales and models; it does not rely on any vendor or third-party software (except for some of the base libraries, noticeably the Java Topology Suite and the JDBC drivers), thus leaving us the maximum programming freedom.

3.4 Conclusions

In this chapter the background of this thesis has been explained: a brief overview of the situation of cartography in Italy was given, highlighting how it is evolving and how the CARGEN project could take part in this evolution. Also it has been explained how the problem of generalization was approached and which were the first design choices; as a whole, the process shaped in the CARGEN project:

has a pragmatic approach to the problem, trying to exploit at best existing solutions, developing new ones only if needed,

relies on the customization of the tools on our specific input and output models and scales,

is developed as a base layer of common functions and a set of modules, each of them handling a specific generalization task.

In the next three chapters the generalization process that has been set up will be explained in details. In particular, chapter 4 will explain the overall process, chapter 5 the model generalization and chapter 6 the generalization algorithms developed.

Chapter 4

The overall generalization process

This chapter will illustrate the overall generalization process that was set up in the CARGEN project. The concept of generalization as the sum of model and cartographic generalization is discussed and the two terms semantic and geometric generalization are introduced; following some relevant peculiarities of the project are highlighted. Finally the description of the overall process is given, modeled as an ordered sequence of steps; for each step the reasons of its position in the sequence is presented. Further details on the implementation of the process will be given in the next two chapters on the semantic generalization and the geometric generalization.

4.1 Generalization process

Cartographic generalization is usually divided into two tasks: model-oriented generalization and graphic-oriented generalization. According to [Gruenrich, 1985], model-oriented generalization takes place when the result of generalization is a geographical database (generalization from primary DLM to secondary DLM) while graphic-oriented generalization takes place when the result of generalization is a map (generalization from DLM to DCM).

In the case of the DB25, the process of generalization should be classified as a model-oriented generalization, since the DB25 is not intended to be used directly to print a map. Nevertheless the IGMI specifications for the DB25 contain requirements also on the representation of the data (e.g. “the number of silos in a group should be reduced if they are too close together”) that bring this product halfway between a DLM and a DCM, requiring both a model-oriented and graphic-oriented generalization.

In this thesis we will use the terms semantic and geometric generalization to indicate respectively the former and the latter operations in this particular context.

The first operation handles the translation of the semantic information from the source data model to the target data model, i.e. how the data present in the tables and attributes of the GeoDBR should be re-classified and stored in the tables and attributes of the DB25.

The second operation handles the transformation of the geometric information of the source data: source geometries should be transformed either to comply with the target data model (e.g. an area in the GeoDBR becoming a point in the DB25) or to comply with some specifications (e.g. two buildings should be merged if closer than 2.5 meters).

The distinction between semantic and geometric generalization is reflected in the design of the generalization process: the first task performed is the geometric generalization, followed by the semantic generalization. This choice lets the geometric generalization algorithms to operate with as much of the original data as possible, allowing them to access the more detailed information of the larger scale database and also to prepare the data for the following semantic generalization (e.g. performing data enrichment).

The generalization process developed is customized for our purpose: the generalization of the IGMI 1:25000 geographical database from the GeoDBR 1:5000 geographical database; this let us exploit some simplification deriving from the scales and the type of generalization involved in the process.

4.1.1 A small gap between large scales

The difference between our input and output scales is not very large: although it is enough to require generalization, the scale gap is small enough for the two models to have a number of similarities. In particular we found a good compatibility between the two data models as most of the feature classes in one are present also in the other and are directly derivable. Working with similar scales meant also that generalization required only modest transformation of the geometries.

On the other hand, as both the 1:5000 and 1:25000 scale can be considered large-medium scales, we had to deal with very rich data models (each comprising more than 200 feature classes) that made the analysis of the model generalization process quite demanding. Furthermore, we found out that not many research works dealt with generalization at such large scales: as existing solutions usually suit a different scale range (1:50000, 1:100000), in some cases we had to develop our new solutions (e.g. the generalization of road junctions) when it was not possible to adapt existing ones (e.g. the simplification of buildings).

4.1.2 DB to DB generalization

Despite the generalization of paper maps and of geographical databases are similar, there are some subtle differences in these processes. When generalizing paper maps the focus is to obtain a good representation of the input data at the target scale; due to representation needs, some of the original data could lose its shape, its original position (e.g. displaced) or be completely lost (e.g. covered by other data, as a label). Despite the errors introduced in the data could be much bigger than the tolerance intrinsic to the target scale (e.g. a road could be displaced much further than only the size of its symbol), these are not considered mistakes if they are functional to obtain a good representation.

On the other hand, when generalizing geographical databases the first concern is the accuracy and correctness of the data. Data is not displaced, nor covered by labels or by the symbolization of nearby objects. Geographical databases try to retain most of the accuracy of the source data, being not readily usable to print a generalized map.

For the reasons above, some of the topics that are typical of cartographic generalization, as displacement and label placement, are not present in the overall process developed, because they are not needed in the framework of our research. Because of the large scale source data, instead, we had to put many efforts in the development of algorithms to remove the details comprised in the source geometries; this meant that the most relevant algorithms focus on the pruning of networks (e.g. roads) and the simplification of buildings. Furthermore, despite the detailed large scale source data and the similarities between the input and output data models, in some cases the source data did not provide all the information needed for the generalization: these situations were solved resorting to data enrichment.

4.2 Putting all together

The generalization process was implemented as a sequence of steps, each of them comprised of a set of algorithms addressing a specific part of the generalization.

4.2.1 Generalization steps

The overall process is composed by ten main generalization steps.

The steps have to be processed in a sequence and every step acts like a black-box: there is no interaction among the steps except from the output of one step being the input of the following; from this point of view the whole process can be seen as a batch process. Each step performs the generalization on a specific type of data: during each step part of the input data is processed and the original source data is gradually generalized step after step.

As the various steps can not communicate among them except by input and output, the order in which the steps are executed is very important. The order has been defined on the base of the importance -according to IGMI specifications- of the data generalized by each step. For example rivers are deemed to be the most important feature class and so they are the first to be generalized: their generalization then is performed on the original data and does not depend on the generalization of any other feature class. Also the dependencies between the various generalization steps had a key role in the definition of the order: each step prepares the data for those following, for example adding enriched information. The dependencies between the steps are illustrated in Figure 4.

In general all the steps concur to prepare the data for the last step, which is the population of the target database. The process flow executes the following steps:

1. generalization of hydrography
2. amalgamation of buildings
3. generalization of the road network
4. generalization of railroads
5. generalization of buildings
6. generalization of ditches
7. generalization of linear features
8. generalization of large areas
9. generalization of points
10. population of the target database

4.2.2 Generalization algorithms

Each step is composed by many algorithms. The algorithms too are run in a sequence, even though the organization is not as rigid as that of the main steps of the process: algorithms can communicate and co-operate to obtain a better generalization. Algorithms can, in some cases, trigger the execution of algorithms that are part of other steps of the process, even though they can not control them during their execution (e.g. the algorithm processing the woods may call the road processing algorithm to build the strokes on the road).

The algorithms usually generalize a single feature class but are aware of the surrounding elements and gather information also from other feature classes of the database.

The algorithms implement different generalization strategies: some of them use a simple condition-action approach, derived from the IGMI specifications (e.g. "all huts smaller than 50 sqm should be deleted"), but most of them use more complex approaches usually comprising a phase of analysis and data enrichment that allows the algorithm to "understand better" the type of object it is working on and to become "aware" of the neighboring objects in relation with it.

4.2.3 Quality controls

In every process, the evaluation of the results and quality controls play an important role. The generalization process designed does not explicitly list any result evaluation step; this however does not mean they are not present. Inside each step the generalization algorithms implement different strategies to assess the quality of the results that they produce and guarantee the correctness of the generalized data. The quality control is then delegated to the algorithms: a description of how each of them handles this problem can be found in chapter 6, while in chapter 7 the quality of generalization is discussed in more broad terms.

4.3 The generalization steps

The list of steps comprising the generalization process is given below; for each step it is explained the reasons behind its position in the sequence, and the algorithms that actually perform the generalization step are listed.

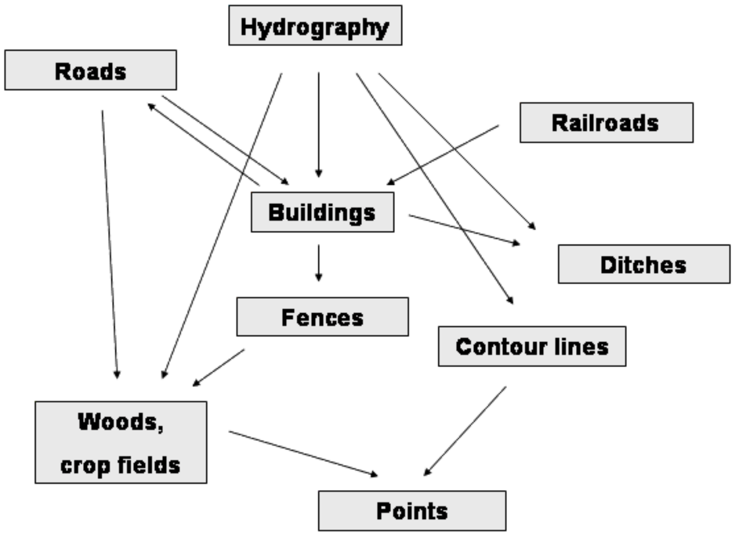


Figure 4: generalization steps and their dependencies; an arrow from A to B means that A influences the generalization of B

4.3.1 Generalization of hydrography

measure of rivers width
collapse of narrow rivers to their midline
harmonization of river boundaries in proximity of collapsed rivers
data enrichment of the rivers
simplification of river boundaries and weeding
pruning of rivers on the base of minimum length
pruning of rivers on the base of density

The generalization of hydrography is the first step of the process: in this way it is not influenced by the generalization of any other feature. During this step rivers are also re-classified on their width.

4.3.2 Amalgamation and selection of buildings

Buildings are generalized in two steps. In the first one the simpler operations are performed: adjacent buildings are merged together and those smaller than a threshold and isolated are deleted. This allows reducing the total number of buildings to elaborate in the following steps.

4.3.3 Generalization of the road network

generalization of highways
identification of dual carriageways, toll-plazas, rest-areas and slip roads
collapsing of dual carriageways

generalization of toll-plazas and rest-areas
generalization of road junctions
harmonization of road classification
line simplification of roads
pruning of the road network on the base of minimum length

Roads, together with rivers and buildings, are the most important feature classes of the dataset. They are processed after the selection of buildings because to prune the network it is necessary to check whether a road candidate to deletion provides exclusive access to any building: having previously merged adjacent buildings and deleted some of them allows for a more correct evaluation of this condition. The generalization of the highways allows to derive four feature classes that are not directly derivable from the source data model.

4.3.4 Generalization of railroads

Railroads are generalized before the buildings because they are used, along with roads and rivers, to partition the space into tiles that might be then analyzed and processed separately (e.g. to aggregate buildings).

4.3.5 Generalization of buildings

aggregation
simplification
pattern recognition and typification

The second generalization step on buildings is executed after all the networks (rivers, roads, railroads) have been generalized. This is because these networks are used to divide the space into partitions and building aggregation is then performed on each partition separately, to avoid to aggregate buildings that are actually separated by a road, a river or a railroad.

4.3.6 Generalization of ditches

pattern recognition and identification of clusters of ditches
typification of cluster of ditches

Ditches are not part of the hydrography network as they do not belong to the graph; furthermore ditches are generalized using typification, while hydrography in general is generalized by selection (pruning). Ditches are generalized after buildings because the typification operator relies on the position of buildings to create the typified geometries.

4.3.7 Generalization of linear features

simplification and collapse of parallel lines
handling contour lines
fences and walls

Fences and walls need to follow the generalization of buildings as their selection relies on the analysis of the content of the area that they surround. Contour lines are processed after rivers in order to adapt their generalization to the paths of the latter.

4.3.8 Generalization of large areas

simplification and aggregation
extension to linear boundaries
collapse to line

This generalization step edits the geometry of large areas, usually representing natural features as wood patches, lakes or crop fields. The IGMI specifications require that the boundary of some of these natural features should be extended to nearby roads, rivers or fences: for this reason they are generalized only after these feature classes have been processed.

4.3.9 Generalization of points

The simplest type of geometry, points are the last geometries to be processed. The generalization of elevation spots and trees should follow that of contour lines and woods respectively as their selection depends on their position.

4.3.10 Population of the target database

selection on the base of the specifications
translation of semantic data

The population of the target database is the last operation that is performed: all the previous steps concurred in preparing the data for this step, enriching the data with the information needed to perform the selection and the translation of the semantic data.

The process has been designed according to the IGMI specifications, and satisfies all their requirements. The design sets a specific order in the execution of the generalization steps that guarantees that all the dependencies between the feature classes are resolved. The generalization process in some cases goes even a bit further than the IGMI requirements, in order to perform a better generalization.

At present day not all the process has been completely developed: the steps are in different moments of the development cycle; some of the algorithms have been fully developed and tested, while other are still being implemented. In particular, some of the algorithms dealing with geometric generalization are under development, while all the steps comprising the model generalization process have been completed. The model generalization process is the topic of the next chapter, while chapter 6 will explain all the steps of the geometric generalization process, providing a detailed description of the most relevant algorithms that have been developed inside this research work.

4.4 Conclusions

In this chapter the overall process of generalization set up for the CARGEN project has been explained. In the process it is possible to identify semantic and geometric generalization, where the first translates the semantic data from the input data model to the output data model and the second transforms the geometric data in order to make it suit the output specifications. It was shown how the process has been modeled in a sequence of steps that are executed in a precise order. Each step comprises a set of algorithms that have been developed to solve a specific generalization problem of one or more feature classes. All the steps concur in preparing the data for the final step of the process, the population of the target database.

Chapter 5

Model generalization

In cartographic generalization, model generalization is the process that translates the content of the source database according to the data model of the target database.

When producing a map, the cartographer abstracts a model of the reality, in which only some of the real world objects are represented, while other are not, as they are deemed to be not relevant to the purpose of the map. In a geographical database, this model is called data model, and defines which real world objects should be present in the database (the feature classes) and which of their characteristic should be stored (the attributes). Two maps at different scales usually adopt two different data models: in fact not all the phenomena that can be shown at the larger scale can be shown at the smaller one, thus leading to the use of two different models to represent the same reality. Because of this, to perform cartographic generalization it is necessary not only to transform the representation of the map objects to adapt it to the target scale, but also to “translate” the semantic data to the target data model: this process is called model(-oriented) or semantic generalization (see chapter 4).

This process can be straightforward if the target feature classes are exactly a subset of those in the source model, that is if every target feature class has a 1:1 correspondence with one source feature class; in any other case it is necessary to operate some transformation on the source semantic data to generalize them.

As at smaller scale less phenomena are visible, it is common that the target feature classes are in a 1:n relation with n source feature classes, that is, a number n of detailed source feature classes will be generalized into a single target feature class, losing their specificity. The reverse case can also be possible, with 1 source feature class originating more than one target feature class. In both cases, the semantic generalization uses attribute values -or some other forms of constraint- to decide how to translate the data from the source to the target feature classes.

It might also be possible that one source feature class has no correspondence in the target model (i.e. what it represents has been deemed not relevant at the target scale); on the reverse, also a target feature class might have no correspondence in the source model (i.e. it represents a phenomena that is not present at the source scale). In the first case, the source data will be lost and will not be present in the target data. In the latter case the missing data might be inferred from other source feature classes, otherwise there is a compatibility issue between the two models and the target feature class will remain empty (this situation could be solved acquiring the missing data from another data source).

Of course these same considerations apply also to the attributes and attribute values comprising the two data models.

Each data model might comprise not only the definition of the feature classes and their attributes, but also some specifications on the data, as size constraints, spatial relations or local condition; also these specifications should be considered during the process of model generalization as they rule how the feature classes should be translated from one model to the other (e.g. a size threshold on a building could decide whether the building should be represented as a point or as a polygon).

In general model generalization requires a bigger translation effort the bigger is the gap between the source and the target scale and the compatibility between the models increases as the purposes of source and target are similar.

In our case the two data models, the source GeoDBR in 1:5000 scale and the target DB25 in 1:25000 scale show a good degree of compatibility and most of the feature classes are derivable.

Despite the similarities, though, the generalization process required developing some algorithms to derive the feature class of the DB25 from the source data; moreover, the two models had some severe incompatibility that required changing the two data models in order to guarantee the derivability of all the feature classes.

In the CARGEN project, the model generalization process was divided in two tasks: the first is the matching, the second is the rule building; both these processes were performed manually. The final result of the process was a Java code that could copy the data from the input database to the target database, performing both the semantic translation and the transformations needed.

Although the description of the data models could provide a detailed background to contextualize the model generalization process and a precise account of the processes of matching and rule building could provide a solid evidence of how the task was demanding and complex, this would probably go beyond the scope of this thesis and add very little from the point of view of the research. This chapter will focus on the most relevant aspects of both the data models and the model generalization process while further information on these topics can be found in [CARGEN, 2009, pp8-208], [IGMI, 2006], [Regione Veneto, 2009]. In the following sections the GeoDBR and DB25 models will be outlined, highlighting differences and similarities between the two; following the process of matching and rule building will be described. The chapter is closed by same examples of the algorithms that were necessary to develop to perform the semantic generalization: the actual explanations of the algorithms are given in the next chapter.

5.1 Data Models

5.1.1 DB25

The DB25 data model is the official IGMI model for the 1:25000 scale.

The model was designed by the IGMI for the new DB25 series maps and it is meant to be derivable from the Regional cartography, although some pre-processing and data integration might be necessary for a complete derivation. The DB25 model is not meant for the direct production of the DB25 maps: it actually represents a DLM, from which the DCM and the maps can be produced with further elaboration; for this reason it tries to retain as much the accuracy of the Regional cartography as possible.

Most of the features are represented with a point or a line; only few of them have a polygonal geometry, noticeably buildings and natural features that extend on large surfaces as lakes, wide rivers, rocks, woods and crop fields.

Networks are represented only using edges and there is not any explicit graph structure.

Most of the features have an acquisition limit, i.e. a minimum size threshold, that determines whether an object should be in the database or not depending on its size; size constraints are also used to classify the same real world object in two different feature classes.

The model comprises 149 feature classes, each of them with a name and a code composed of one letter indicating the geometry type (A: area, L: line, P: point) and the FACC code [DGIWG, 2000].

Inside each feature class, the IGMI model describes one or more objects, that we will call Labels. Each Label is a particular instance of a feature class and represents exactly one type of map-object. Each Label has an unique identifier, stored in the attribute “LAB” (label), has its own definition and its own specifications (that usually are inherited by the feature class it belongs to).

The total number of distinct Labels is 239; since in the IGMI model the Labels represent the actual objects that store the information, the model generalization process focused on the derivation of each different Label from the GeoDBR model.

5.1.2 GeoDBR

The GeoDBR data model is the Regional model of Regione Veneto for the 1:5000 and 1:10000 scale. As the definition of a national data model for the large scale is not yet complete, the GeoDBR is slightly different from the most recently proposed national model; nevertheless these two models are quite similar and what is presented in this thesis can be easily applied to the developing National model.

Because of the large scale few acquisition limits are given in the GeoDBR: almost every object in the data model is inserted in the database despite its size. Most of the features have a polygonal geometry, except those that in reality resemble very closely a line, as very narrow rivers, ditches, power lines, cableways or pipelines. Points are used to represent very small objects, as poles or to mark special points on the terrain as springs, the entrance to a mine or elevation spots.

The networks of roads, railroads and hydrography are represented using polygons (except the narrowest rivers, represented as lines). These networks are also represented as a graph: the road network (highways included), the hydrography network (both natural and artificial streams, ditches excluded) and the railroad network are represented in a node-edge structure. As a design choice, the attributes of these features are stored in the edges of the graph, while the geometries (polygons or lines) are used to represent the extent of the features. The graph edges have also an actual geometry, representing the middle line of the feature. There exist a 1:1 relation between each edge and the feature it carries the attributes of: for this purpose features are divided into pieces that correspond to each edge of the graph.

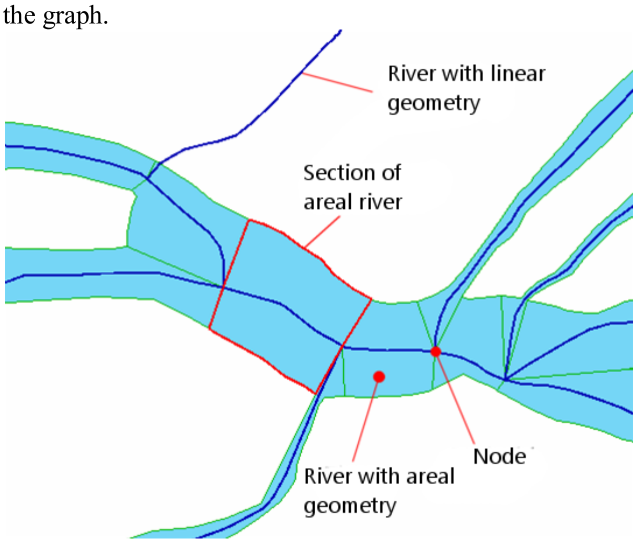


Figure 5: example of the feature classes representing the hydrography

The existence of the graph on one hand divided the source features in many pieces, requiring us to develop algorithm able to put together all the pieces to gather some global information (e.g. to calculate the length of a whole river); on the other hand it made unnecessary to collapse to line features like rivers and roads as the middle line of the features could be retrieved from the graph.

5.2 Matching

The process of matching, that is to find the correspondences among the feature classes of the source and target models, required to study carefully the specifications of both the data models and to find the correspondences by looking at the definitions of the feature classes, the attribute and the attribute values in the two data models.

During the study we found a critical situation about the model of roads: the classification of roads in the DB25 relied on information that was not present in the GeoDBR, in particular the road surface material; as such, the DB25 feature class LAP030 (Road) could not be derived from the GeoDBR, hindering the whole process. To solve this incompatibility, we proposed to IGMI to modify its data model, bringing it closer to the standard that will be used in the national data model (roads are classified by importance using a numeric attribute, whereas the DB25 classification of roads relied on a set of attributes, among which road surface material, sometimes with overlapping definitions). The revision of the road modeling affected also that of tunnels and bridges that to the former were related.

Also other Labels could not be derived from the source GeoDBR data: in some cases we proposed the IGMI to drop the Label, as it represented an object of minor importance (e.g. trough), in other cases we proposed Regione Veneto to add the object to its model.

In general, the matching process allowed to test the compatibility between the two data models and to improve it; as the modifications of the models were received and accepted by IGMI and Regione Veneto, the DB25 was completely derivable from the GeoDBR.

At the end of the matching process, we identified three main groups of DB25 Labels:

Labels directly derivable
Labels derivable but subject to some specifications
Labels not directly derivable

5.2.1 Labels directly derivable

This type of Labels can be derived by simply using SQL queries, with no further processing. This means that there is a very good match between the GeoDBR and the DB25 data models on the object described by this Label: the geometries in the DB25 are the same of the GeoDBR and semantic data needs only some minor adjustments (e.g. to change an attribute value).

5.2.2 Labels not directly derivable

Despite the two data models have been aligned, some Labels of the DB25 do not have any match among the feature classes of the GeoDBR; to derive these Labels it was necessary to gather the data processing the source data. In some cases it was sufficient to apply a spatial operator (e.g. see Figure 6); in other cases complex procedures of data enrichment had to be developed (e.g. see the classification of highways in chapter 6)

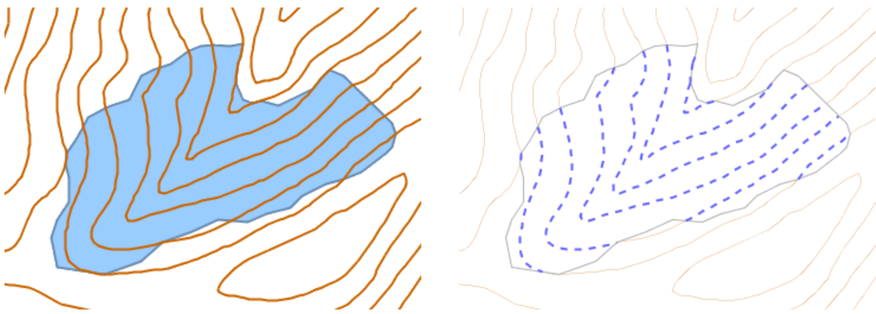


Figure 6: an example of Label not directly derivable. The Label "Contour line on glacier" has no match in the GeoDBR. To derive this Label is necessary to select the intersection between the GeoDBR feature classes "Glacier" and "Contour line". Left: initial data (contour lines in brown, glacier in light blue). Right: derived data (contour lines on glacier are dotted).

5.2.3 Labels subject to specifications

Specifications decide whether and how a feature of the GeoDBR should be stored in a Label. There are four main types of specifications:

acquisition limits
pre-processing requirements
generalization rules
other

A Label can be subject to one or more of these four types of specifications; the specifications work as constraints: only if all of them are satisfied the source feature will be generalized. While Labels with no specifications could be directly generalized from the source data using simple SQL queries, those with specifications required some processing to be generalized, in some cases leading to the development of ad hoc algorithms.

Acquisition limits specifications

There are two types of these specifications:

geometric constraints
spatial constraints

Geometric constraints set a minimum size threshold for the objects. The thresholds could be on the width, length, height or area size of an object. Length and area size are easily evaluated, while the evaluation of width and height required to develop two different algorithms.

Spatial constraint rule whether an object should be generalized or not depending on the presence or absence of other objects in its surroundings. For example the specifications require that for a mountain pass to be classified as the

Label P320 of the feature class PDB150 Mountain Pass it should be in proximity of a minor road.

The evaluation of these constraints required to develop specific algorithms.

Acquisition limits specifications could be also mixed: e.g. the Label P403 of the feature class PAL100 Hut are present in the DB25 only if they are isolated and have an area bigger than 50 sqm.

Pre-processing specifications.

They require that some operations should be applied to the objects before the generalization. There are just two different type of pre-processing specifications:

aggregation of areas
extension of lines

The first states that groups of similar objects (e.g. crop fields) should be aggregated if closer than a certain distance threshold, while the second states that gaps under a certain threshold in linear objects (e.g. fences, see Figure 7) should be ignored and the object should be generalized as a continuous line.

The specifications require that these operations should be performed before the actual model generalization, as the acquisition limits should be evaluated on the new aggregated or elongated objects.

Figure 7: fences (black lines) in a urban context (buildings in yellow, roads in brown); according to IGMI specifications, a gap smaller than 10 m in a fence should be ignored and the fence derived as a continuous line

Generalization rules

Since the DB25 data model was developed with the aim to be derivable from the Regional maps, it contained also some specifications on how to generalize some of the Labels. This kind of specifications will state, for example, that the Labels L626 of the feature class LBH030 (ditches) in the presence of a high density of the same features, should be generalized taking in consideration only those further than 100 m from each other.

These generalization specifications are at the base of many of the generalization algorithms explained in the next chapter.

Other specifications

Some Labels had some special specifications on how to derive them from Regional maps. In general these specifications had to be analyzed one by one by hand and solved developing a specific solution.

5.3 Rule building

The process of rule building translates all the relations among the feature classes of the input and output models into a set of formal rules that are used to develop the Java code that performs the model generalization.

Working with databases, we decided to use SQL as the language to formalize the rules; since not all the rules could be expressed using only SQL commands, we used an extended notation, adding some custom commands to indicate special functions that needed to be applied to comply with the specifications. The list of these custom commands, along with their explanations and examples, is found at the end of this chapter.

To speed up the creation of the rules a special tool was developed. This tool allows the user to pick a Label from the DB25 model and to pick a corresponding feature class of the GeoDBR model and its attribute values, thus creating a mapping rule between a Label and a particular instance of the a feature class of the GeoDBR. The user can then define how to populate the attributes of the DB25 Label, either typing the values for fixed values attributes, or writing an expression that maps exactly the relation between the attribute values of the GeoDBR and DB25. For this purpose, a simple scripting language was developed: this language allows the creation of simple SET-IF statements, to embed in the rule the acquisition limit specifications and, using some custom codes, also some of the generalization specifications. A screenshot of this tool can be seen in Figure 8.

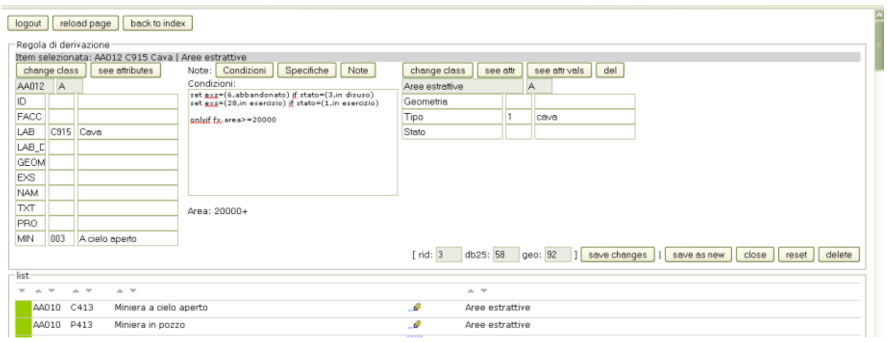


Figure 8: screenshot of the tool for rule building: on the left the attributes of the DB25 feature class, on the right that of the GeoDBR; in the middle the space to express eventual specifications with the scripting language.

Once all the rules have been created, the tool can output them either as a complete report (this very same functionality was used to create the CARGEN documentation), or as a list of queries written in the extended SQL notation. The queries are then embedded into a Java code that performs the model generalization, actually populating the tables of the generalized database. The queries of the Labels with no specifications need few coding, as they can be directly sent to the server via JDBC and executed; instead the queries containing special functions are transformed into a more complex Java code.

As it was explained in chapter 4, the semantic generalization process is executed after the geometric generalization: pre-processing specifications, acquisition limits on width threshold and generalization specifications are all handled during the geometric generalization process; from this perspective, the process of model generalization can not be considered isolated, as it blends in the overall process.

5.3.1 Custom extended SQL notation

The list of custom commands used in the extended SQL notation is given below; some of these commands are automatically translated by the rule building tool to valid SQL statements (e.g. in PostGIS or Oracle Spatial notation), other need to be translated by hand into Java algorithms.

FX.LEN

Function to measure the length of a geometry; can be directly translated to a call to SDO_GEOM.SDO_LENGTH() in Oracle Spatial, or to ST_LENGTH() in PostGIS.

FX.H

Function to measure the height of a geometry; this has been implemented as the difference between the highest and the lowest Z values of the vertices of the geometry.

FX.W

Function to measure the width of an areal geometry. The width of a polygon can be difficult to formalize, and there are many different ways to evaluate this measure. Our approach was to compute the distance between a line running in the middle of the polygon and the boundary, drawing a line orthogonal to the center line and measuring the distance between the points of intersection with the boundary.

FX.AREA

Function to measure the area of a polygon; can be directly translated to a call to SDO_GEOM.SDO_AREA() in Oracle Spatial, or to ST_AREA() in PostGIS.

SQ()

Command that means that it is necessary to run a spatial query. The text between the brackets explains in detail how the spatial query should be performed.

FX.GEOM(geom_a > geom_b)

Command that means that the source and target feature classes use two different types of geometry, and thus it is necessary to implement a function to generalize the latter from the former. The text between brackets might be one of the following:

FX.GEOM(centroid)

Function that returns the centroid of the input feature; can be directly translated to a call to SDO_GEOM.SDO_CENTROID() in Oracle Spatial or to ST_CENTROID() in PostGIS.

FX.GEOM(axis)

FX.GEOM(asse contenuto)

FX.GEOM(asse ferroviario contenuto)

FX.GEOM(asse stradale contenuto)

Function that given an input feature will return the corresponding edge of the graph associated to its feature class. If the feature class of the input feature does not have any corresponding graph (i.e. is not either a road, river or railroad segment), the output of the function FX.GEOM(medial) is returned.

FX.GEOM(medial)

Function that given an input polygon will return its center line; this function performs an area to line collapse.

FX.GEOM(boundary)

Function that returns the perimeter of a polygon; can be directly translated to a call SDO_UTIL.SDO_POLYGONTOLINE() in Oracle Spatial, or to ST_BOUNDARY() in PostGIS.

FX.GEOM(sides)

Function that returns the perimeter of a polygon, but if the element touches elements of the same feature class, from the perimeter are subtracted the parts that are in common with the boundary of the neighboring features.

FX.GEOM(head)

Function that given an input polygon, will return the highest part of the boundary, meant as the sequence of consecutive points having Z value higher than the average Z value.

FX.GEOM(corresponding geometry)

Function that is the inverse of FX.GEOM(axis): given one edge of a graph will return the geometry of the corresponding feature.

5.3.2 Some examples

To better understand the use of the extended SQL notation, some examples are given below. The examples are an extract from the document [CARGEN, 2009]; the queries highlight on one side how the complexity of the correspondences between different models require human intervention to be solved, on the other how this operation is dependent to previous geometric generalization operations.

```
INSERT INTO LAP050 ( FACC,GEOMETRY,LAB,LAB_DESC ) SELECT 'AP050',FX.GEOM( ASSE ),'L715','Vialetti parchi/giardini' FROM AC_PED WHERE SQ(DENTRO PARCHI/GIARDINI)
```

This query populates the Label L715 Vialetti parchi/giardini of the DB25 feature class LAP050 Trail/Footpath using the GeoDBR feature class AC_PED (pedestrian area); from the source data the query reads only the geometry and no other attributes are used. Since AC_PED is represented by polygon geometries, while LAP050 stores lines, the query embeds a call to the function FX.GEOM(ASSE); furthermore, to follow the IGMI definition of the Label, it is necessary to perform a spatial query to select only those element of AC_PED being inside a park or garden (this is indicated by the argument of the SQ() command).

```
INSERT INTO LAQ040 ( FACC,GEOMETRY,LAB,LAB_DESC,BSC ) SELECT 'AQ040',FX.GEOM(ASSE STRADALE CONTENUTO),'LX22','Ponte/Viadotto per autostrade','014' FROM PONTE WHERE FX.LEN>=2 AND SQ(ASSE=AUTOSTRADA)
```

This query populates the Label LX22 Ponte/Viadotto per autostrade of the DB25 feature class LAQ040 Bridge/Overpass/Viaduct using the GeoDBR feature class PONTE (bridge). In the DB25 model different Labels are used in the LAQ040 feature class to distinct the type or road passing over a bridge: in the example the Label LX22 represents highway bridges. To populate each Label is then necessary to perform a spatial query to select only the elements in PONTE containing one or more edges of the road graph classified as highway (this is indicated by the argument of the SQ() command). To assure a perfect match between the line representing the bridge and the road in the DB25, the function FX.GEOM() will not compute the center line of the polygon geometries in PONTE but instead use the edges found by the spatial query (this is indicated by 'ASSE STRADALE CONTENUTO'); this of course requires that the geometries representing the highway have already been collapsed to a single center line.

5.4 Conclusions

This chapter described how the model generalization was approached: the two data models involved in the process, the DB25 and the GeoDBR, are briefly described, highlighting differences and similarities among the two. Then the process of matching and rule building are illustrated: the former revealed some incompatibilities between the source and target data models that had to be solved; the latter required to develop an extended SQL notation and to implement a tool to ease the creation of the semantic generalization rules.

Chapter 6

Generalization algorithms

In this chapter the most important algorithms to perform the generalization of the DB25 will be described.

The main purpose of the algorithms presented in this chapter is to transform the original geometries of the features of the GeoDBR in order to make them suit the DB25 specifications and data model; although these algorithms focus mainly on the geometric aspect of the features, it will be shown how they also rely on semantic data and in some cases enrich the semantic data with information gathered from the analysis of the geometries. Following the explanation of all the algorithms, a brief outline of all of them, classified as operators, is given. The chapter is closed by some final remarks on the most important operators and the possibility to use the algorithms to generalize also smaller scales.

In generalization, the development of the procedures to transform the geometries of the features -the so called “operators”- is surely the most challenging task. The generalization of the objects on a map requires a set of skills that a computer does not natively possess and that need to be taught to it. Somehow, it is necessary to teach the computer how to draw a map.

If this objective is probably too ambitious as a whole, it is possible though to develop generalization algorithms if they focus on small and specific traits of the generalization process: a specific input and output scale, a specific input and output model and a specific problem to solve.

All the algorithms presented in this chapter have been developed following this approach: every algorithm developed solves a specific generalization problem; sets of algorithms have been grouped together to generalize a specific set of feature classes. In particular, this chapter will describe the algorithms for the generalization of:

the hydrography network
ditches
the road network
the highway network
small regular areas as buildings
big irregular areas as wood patches and crops

lines as pipelines or contour lines
points

As the research in the CARGEN project goes further, new algorithms are being developed and old ones are being improved; in the following pages the present state of the development is presented, with the explanation of the implemented algorithms along with the description of those algorithms that have been designed but not yet developed.

6.1 Generalization of buildings3

Buildings, together with roads and rivers, are one of the most important features in a map; buildings are related to the presence of man and their presence or absence represent a valuable information in a map: for example a single building can provide shelter for a trekker and a group of buildings can tell to a merchant where a settlement is. Having such a central role in cartography, buildings have also received lots of special attention in the context of generalization [Regnauld and McMaster, 2007].

6.1.1 Related work

Many different algorithms have been developed to generalize buildings, as their representation changes a lot at different scales: at larger scales buildings are still represented as single objects while at small scale all the buildings in a city could have been merged together in a single geometry that bears no memory of each individual object. At larger scales, when buildings are still treated as single entities, they can be simplified by removing the smallest details [Sester, 2000], [Haunert and Wolf, 2008], [Fan and Meng, 2010] or replacing each building with a simplified version of itself through template matching [Revell, 2005], [Rainsford and Mackaness, 2002]. As buildings are usually found grouped into settlements, many algorithms deal with groups or cluster of buildings [Sester and Brenner, 2000]. When the scale decreases, buildings in groups can be deleted or merged together [Regnauld, 2003], [Li et al., 2004] or be typified, that is reducing the number of buildings in the cluster trying to maintain their original spatial distribution [Regnauld, 2001], [Burghardt and Cecconi, 2003]. Buildings and roads are closely related to each other: roads for instance can be used to divide buildings into groups (e.g. see [Agent, 2000]) and the buildings, when generalized, should maintain their orientation with the road [Chrisophe and Ruas, 2002]. The problem of displacement of roads and buildings have been long studied and

[3] This work was done also with Damiano Callegari, University of Padua.

solutions have been proposed by many [Mackaness, 1994], [Ware and Jones, 1998], [Bader and Barrault, 2001].

6.1.2 Specifications

The development of the generalization step started from the analysis of the IGMI specifications for the 1:25000 scale. According to these specifications the requirement on buildings are quite simple as they only define:

a minimum building size (50 sqm),
a minimum building distance (3 m),
a minimum courtyard size (300 sqm).

Displacement was not among the requirements, while typification had to be applied only in the generalization of sets of silos. The process developed consists in seven algorithms, to be executed in a sequence; each algorithm prepares the data for the following step or enforces one of the IGMI requirements on the data. The algorithms developed are explained in detail next.

6.1.3 Selection of buildings

According to IGMI data model, in the generalization between the 1:5000 to the 1:25000 scale, most of the source buildings should be retained. The specifications state that only buildings that have an area size smaller than a threshold value should be deleted, and this should be done only if the building was not isolated. To detect the isolation of the buildings, the algorithm draws a buffer around each small building (i.e. with area smaller than the threshold) and finds whether any other building is inside this buffer: in this case the building is not isolated, otherwise it is. The radius of the buffer was set to 500 meter, a measure that was evaluated to be a good tradeoff between the number of buildings deleted (thus freeing some space) and those retained (useful as a landmark). Any building not isolated and with area below the minimum building size is deleted.

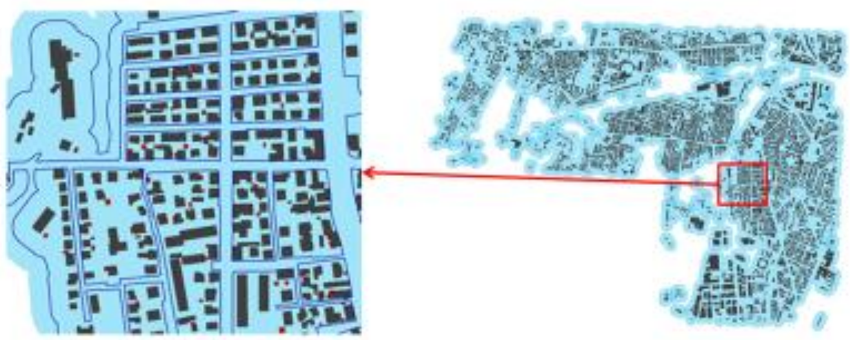


Figure 9: buffers are used to detect isolated buildings; a building is isolated only if its buffer does not contain any other building

6.1.4 Amalgamation

The purpose of the aggregation algorithm is quite trivial: all the buildings that are adjacent should be merged together. The algorithm developed actually is slightly more complex, as it performs two tests before actually merging two adjacent buildings:

the intersection between the two buildings is computed and its size checked: if it is too small the buildings are not amalgamated, but are flagged as to be processed using the aggregation algorithm (explained below);

the aggregation is performed only on buildings that will be then classified in the same DB25 feature class: for this purpose a compatibility function returns true or false whether two adjacent buildings should be aggregated or not.

If two adjacent building pass both these two controls, they are merged together.

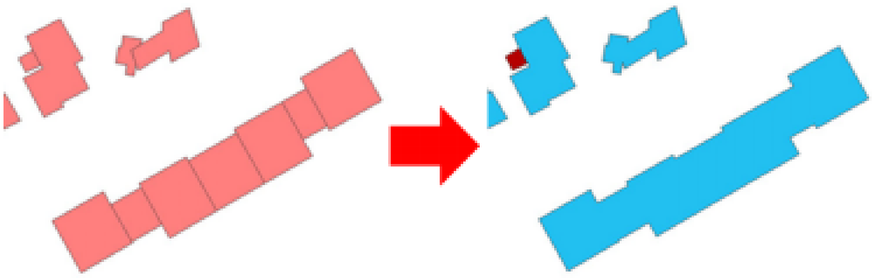


Figure 10: building amalgamation; left: source data, right: after amalgamation

6.1.5 Aggregation

The process of aggregation is used to merge together two buildings that are not adjacent. Since there is some space between the two buildings, the algorithm should find a way to fill the gap. There are mainly two approaches to this purpose: build a new geometry that will cover the blank space and connect the two buildings, or to move the buildings in order to make them adjacent [Regnauld, 2003]. In the first approach the difficulty is to create a new geometry that can be inserted seamlessly between the existing ones, while in the second it might be difficult to evaluate how to move the buildings in order to avoid small gaps in the resulting merged geometry.

Our choice was to develop an algorithm following the first approach that fits best our large scale data, that has dense sets of detailed buildings: the first approach in fact scales more easily to contexts of multiple buildings to be aggregated together, while using the second approach the complex outlines of the buildings could lead easily to small gaps in the merged geometries.

Existing aggregation algorithms use a triangulated mesh to generate the new geometry to connect the disjoint buildings (or objects in general) [Bader, 1997].

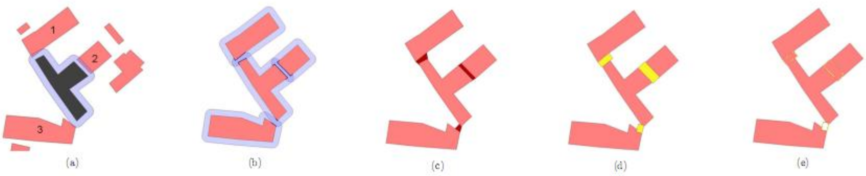


Figure 11: aggregation: a) finding buildings under threshold distance, b) buffers, c) geometry connecting the points of intersection between the buffers, d) MBR of the geometry, e) if the area of the MBR is too small (white MBR) the two buildings are not aggregated

We found that while this approach works well for natural features (crops, woods), it does not suit perfectly the task of aggregating buildings: due to the triangles edges, the shape of the new geometries were found to be too “soft” to fit the generally more angular shape of buildings; moreover the new geometries generated could be very narrow, looking like corridors connecting the buildings.

The solution devised then uses another approach to build the connecting geometry: this is created as the convex hull of all the points of the two buildings that are within a distance threshold from the other building. To solve the problem of how the shape of the new geometry fits among the existing building, the new geometry is made angular computing its oriented minimum bounding rectangle (see squaring). This choice may seem bizarre as it creates geometry that will not “blend” with the surrounding ones; instead this choice is justified as it actually prepares the new geometry to be “smoothed” by the simplification algorithm (explained later) that follows aggregation in the building generalization process.

In details the algorithm works as follows:

1. a buffer of radius R is drawn around each building, where R is the minimum distance set by the IGMI specifications; the intersections among buffers and buildings detect which buildings are under this threshold distance and should be aggregated
2. for each couple of buildings to aggregate A and B, the points of intersection between the buffer A and the building B (and vice versa) are calculated
3. the convex hull of the points of intersection is drawn
4. the size of the area of the convex hull is calculated: if it is too small (under the square of the minimum distance threshold), the two buildings are too far away and they are not aggregated
5. the oriented minimum bounding rectangle of the convex hull and the two buildings A and B are merged in an unique geometry

6.1.6 Simplification

The simplification of the buildings outline is not explicitly required by the IGMI specifications. Nevertheless the inspection of the source geometries in 1:5000 scale revealed that they were too detailed for the target 1:25000 scale; in particular they featured details (i.e. small just in the building facades) that would have been too small, that is below the accuracy of the 1:25000 map that, according to IGMI specifications, it is set to 2.5 meters or 0.1 mm on a the paper map. The source geometries were also composed by a very high number of points

It was our choice then to develop an algorithm to simplify the geometries of the buildings. The simplification strategy relied on two algorithms: one to reduce the number of points in each geometry, the other to remove the small details from the facades. As the generalization of buildings is one of the most studied topics, we could find two existing algorithms that we could use to achieve the simplification.

6.1.6.1 Reduction of vertices

We applied the well known Douglas-Peucker algorithm [Douglas & al., 1973] to reduce the number of points comprising the shape of the buildings. The Douglas-Peucker algorithm is a recursive line simplification algorithm that given an input line and a tolerance value will compute an approximation of the input line that is described by a subset of the points describing the input line and lies at a distance from them smaller than the tolerance.

The idea behind the algorithm is quite simple: to approximate a line *An,An+1,.....An+m* composed by m points the algorithm computes the line *An,An+m* (baseline) and finds the furthest point from this line among the m points. If the distance to the baseline is below the threshold, the line is approximated by the baseline, otherwise the line is split on the furthest point and the algorithm recursively approximates the two pieces.

Douglas-Peucker algorithm is a line simplification algorithm, but it can also be used on polygons applying it to the polygon boundary (in the case of compound polygons or polygons with holes it is necessary to operate on each ring singularly). Douglas-Peucker algorithm is fast even in its base implementation (a faster implementation exists [Hershberger & al., 1992]), especially on polygons with few vertices (a typical building has often less than 20 vertices) and it is able to retain the most characteristic shape of the input line using a small part of the input vertices. Although other line simplification algorithms exist [McMaster, 1987], our choice fell on Douglas-Peucker as it is readily available in many libraries (e.g. JTS) and it is easy to setup (requires only one parameter).

A problem of the base implementation of Douglas-Peucker algorithm is that it is not topological safe: the simplification of a closed line may in fact create self-intersections; a topological safe version exists [Saalfeld, 1999], but it has a higher computational cost. Also, the use of the Douglas-Peucker algorithm on buildings or rectangular-shaped objects is not completely recommendable as it tends to delete the corners, making the remaining very sharp.

Our point reduction strategy applies the Douglas-Peucker algorithm to the building outlines with a very small tolerance: this allows to reduce the number of vertices, although limiting the drawbacks of the algorithm (self-intersecting and rounded outlines): a threshold of 1 meter has been experimentally found a good tradeoff between the number of vertices deleted and the absence of errors.

6.1.6.2 Elimination of juts

To eliminate the smallest details of the facades of the buildings, we based our strategy on an algorithm described by Monika Sester [Sester, 2000]. The algorithm is an iterative procedure that removes from a building all the facades that are shorter than a threshold; the decision of how to remove a short facade depends on the geometry of the neighboring sides. Sester's algorithm handles three distinct cases:

intrusion / extrusion: the angle between the preceding and the subsequent side is approximately 180°: the small side is set back to the level of the main facade.

offset: the angle between the preceding and the subsequent side is approximately 0°: the longer one of the adjacent building sides is extended, and the shorter side is dropped.

corner: the angle between the preceding and the subsequent side is approximately 90°: the adjacent facades are intersected.

These rules are iteratively applied to all the small sides of a building, starting with the shortest ones.

We implemented a modified version of the simple Sester's algorithm described above, in order to adapt it to our data.

First we extended the application of the algorithm by widening the range of angles treated: in the original implementation the algorithm simplifies only building that are almost rectangular (with almost square corners) while in our source data, the building sides are connected to the neighboring sides with angles that are not treated in the original implementation; the range of angles treated has been increased by +-15° on each case.

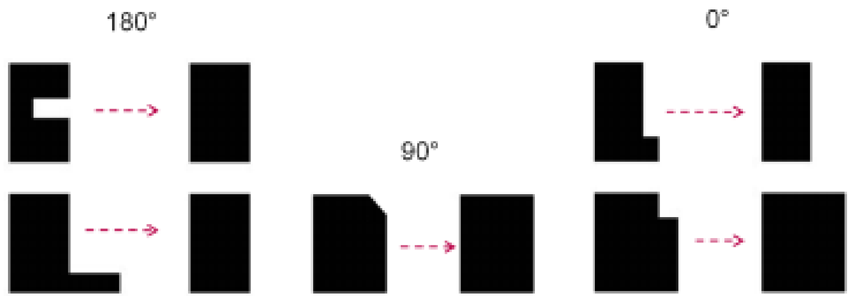


Figure 12: transformation of the building outline according to Sester's algorithm

As a second modification we changed the solving strategy in the “offset” case: our implementation extends the longer side but moves it back, toward the inside of the building, to keep the area of the building constant.

Also the solving strategy of the "intrusion/extrusion" case is changed: in the case the setting back of the small side to the level of the main facade causes an area loss or gain bigger than a threshold, the operation is not performed, but the intrusion or extrusion is exaggerated widening the small side to reach the minimum side size.

The application of the algorithm reduces the number of vertices, and thus of sides, in the buildings; it also simplifies the rectangular shapes created by the aggregation, blending together the buildings that were merged.

The parameters to operate the algorithm have been inferred from the specifications or found empirically by visual inspection of the results: the minimum side size chosen is 3 meters while the maximum area loss or gain has been set equal to the minimum building size of 50 sqm.

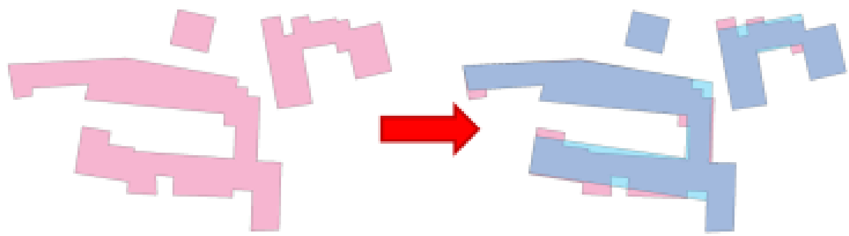


Figure 13: example of the application of the juts elimination algorithm

6.1.7 Squaring

The operation of squaring aims at giving the building a squared look, thus helping the user to identify the building. This is usually necessary to overcome the accuracy limitations of the digitization process [Regnauld & al., 2007]. In practice, at large scale it is not possible to square all the corners of each building, so our implementation aims at reducing the number of different values of the angles of a polygon. The basics of the algorithm are quite simple: a base angle and a polygon are given as inputs to the algorithm and the algorithm will modify each angle of the polygon in order to round its value to the closest multiple of the base angle. The base angle is a fraction of the right angle (e.g. 90/3, 90/4). The effect of the squaring operation is a discretization of the number of allowed angles: this will deform the original shape of the building.

In order to not introduce big deformations to the original shape of the building, the angles should be "squared" referring to the main orientation of the building (e.g. see [Duchêne et al., 2003]): before performing the squaring, the algorithm detects the main orientation of the building. The squaring could lead to an excessive deformation of buildings with many different angles: to avoid this, before performing the squaring, also a "square-ability" test is performed on the building. The three steps comprising the squaring algorithm are explained next.

6.1.7.1 Detection of orientation (calculation of the oriented minimum bounding rectangle)

To detect the main orientation of the building a polling strategy is used: every side of the building casts a vote for its direction and at the end of the votes, the most voted direction is chosen. The directions are measured in module 90; to give bigger importance to longer edges, the votes are weighted (multiplied) by the length of the side that casts the vote. The implementation is quite trivial: an empty array of 90 cells is prepared, then for each side of the building, the direction and its length are measured; the length of the side is then stored in the array in the cell indicated by the module 90 of the direction of the side (rounded to integer). At the end of the operation the cell storing the highest value represents the main direction.

The directions are measured in module 90 to obtain a better chance to detect the main direction: in this way sides that are orthogonal will cast a vote for the same direction that will more easily be picked as that having the more votes; the second main direction is supposed to be orthogonal to the main direction. The module 90 implies that we do not know exactly the main direction of each building, but only that it is either that returned by the algorithm or the one orthogonal to it; however this is enough for the squaring algorithm to run correctly.

The main orientation of a polygon can also be used to compute its orientated minimum bounding rectangle: this is done by finding the main direction R of the polygon, then rotating the whole polygon by an angle $-R$, compute its MBR and then rotating the MBR of an angle $+R$.

6.1.7.2 Square-ability test

As the squaring operation changes the value of the angles, it rotates the sides of the buildings. The rotations are more noticeable if they occur on long sides and, in some cases, they may cause a long side to intersect another side, resulting in a wrong geometry. It's important then that long sides are not rotated: the weighting applied during the detection of the orientation should bias the polling according to this purpose. In the case of a building having too many long edges directed in too many different directions the algorithm is very likely to produce a wrong geometry: as one direction will be picked to set the main orientation, the long edges that are rotated will probably cause self-intersections in the building perimeter. To avoid this problem a square-ability test is run before actually squaring the buildings.

The square-ability test has been devised to detect those buildings that are not fit to be squared.

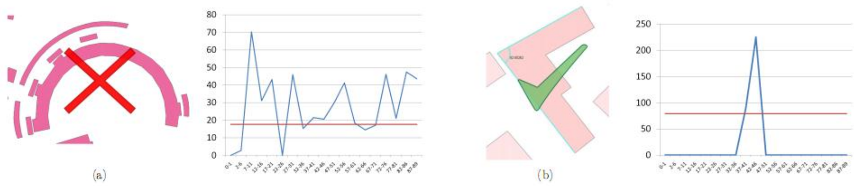


Figure 14: testing two different buildings: a) the algorithm can not find the main direction of the building and it will not be squared; b) the building has a clearly one main direction and it will be squared. In the plot: in red the average, in blue the votes for each direction

The concept behind the test is quite simple: in buildings with many different angles, the values of the direction of the sides will be spread in all the cells of the polling array, while, on the opposite, in a perfectly squared building they will be found only in one cell. The test then works by counting how many cells have a “high” value: if they are more than one, the building does not have only one main direction, but more than one and thus is not fit to be squaring.

To find the “high” values the test computes the mean angle value: this is done counting the sum of all the cells with not zero value and dividing the sum by the length of the perimeter of the building. To detect more easily the “high” values, during the test the angles are grouped in “macro-cells”: they are measured in module 90, but then, through a rounding operation, they are divided onto an array of 90/n cells, where n is the group size; this has the effect to locally minimize the dispersion of the measures. If the number of macro-cells whose value is bigger than the mean angle value is one, the building can be squared, otherwise not.

6.1.7.3 Angle squaring

The algorithm that actually squares the building iteratively rotates each side of the building in order to change its rotation to the closest allowed angle (a multiple of the base angle). Each side is rotated around its centroid to minimize the effect of the rotation on the total area of the building and on the offset of the building. When each side is rotated, the intersections with the previous and following sides are calculated, and the position of the vertices is updated. In the case that adjacent sides are rotated to the same angle, thus becoming parallel, the sides are merged into a single side that is rotated around the centroid of the sides merged.

At the end of the process, the directions of all the sides of the building and its angles have a value that is a multiple of the base angle.

6.1.8 Removal of internal rings and spikes

To abide the IGMI specifications, courtyard among buildings should be removed if their size is smaller than the defined threshold of 300 sqm. This is achieved easily deleting every internal ring of the polygons representing the buildings whose area is below the threshold.

The process of amalgamation, aggregation and simplification could have created some small gaps between the buildings; if these gaps are inside the building perimeter, they are internal rings and are then deleted when removing the courtyards, while if they are on the perimeter of the buildings, they are spikes. Spikes are removed from buildings using a very simple algorithm that detects them by measuring the angle between each two adjacent sides and the length of the same sides. If the angle or the lengths are below a threshold, a spike has been found and it is removed by extending the side adjacent to the shortest of the two sides.

6.1.9 Typification[4]

The operation of typification is a selection operator that tries to maintain spatial patterns. Typification is an operation that is widely used to perform generalization of buildings at small scales. For our purposes, typification is not necessary to generalize buildings; instead IGMI specifications suggest to use it to generalize patterns of silos. This let us to develop a simple algorithm that is run just on the feature class of silos. The algorithm tries to find if a silo is isolated or in a group and if the group has some spatial pattern distribution: in this case it tries to delete some of the silos while at the same time still conveying the information about the spatial pattern.

To detect whether a silo is isolated or in a group, the same procedure to find isolated buildings is used. When a group of silos is found, they are processed by the typification algorithm.

The typification algorithm can recognize linear and grid patterns, with the latter being an extended case of the former. The objective of the algorithm is to delete some of the silos in each group in order to free enough space that each silo is at the minimum distance value from the neighboring ones.

The first step of the algorithm is to test whether there exists a line that passes through all the silos in the group: the centroid of every silo is connected to the centroid of the nearest one by a line segment and the average of the direction of these line segments is computed. Then this average is used to draw a line on each centroid of the group: if at least one of the lines crosses all the silos, they are in a linear pattern, otherwise they are not and the typification algorithm is not applied to them. The line that crosses all the silos in the group is taken as reference (if there are more than one line, that with the smallest average distance from the centroid of the silos is chosen): the algorithm will try iteratively to place the silos equally spaced on this line, deleting one silo at iteration in the case the space is not enough.

In some cases silos are aligned in a grid pattern; this means that they are aligned along two main directions, usually orthogonal one to each other, and that every silo belongs to exactly two linear patterns, each parallel to one main

[4] This work was done with Rossella Baldin, University of Padua.

direction. To generalize grid patterns the idea is to consider them as a 2D extension of a single linear pattern: the algorithm is iteratively applied to generalize the linear patterns along one direction and then along the other direction. Comparing a grid pattern to a matrix, the algorithm will first solve all the rows and then all the columns; each row (or column) is treated as a single linear pattern. The process is iterated until the typification led to a reduction of the number of silos (along both directions) that leaves enough free space between one silo and the neighboring ones.

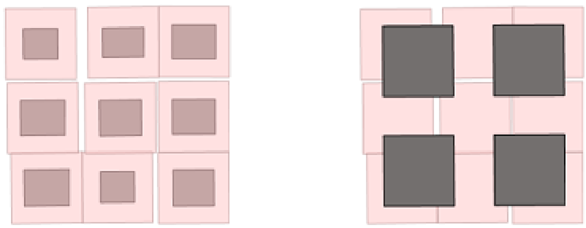


Figure 15: typification of a grid of similar objects

6.2 Generalization of the road network

The road network is one of the most relevant features in a map and probably represents the most recognizable sign of anthropization. The importance of roads, their ubiquity and their relation with other themes make them one of the main topic of generalization.

This chapter will present the algorithms for the generalization of the road network from the scale 1:5000 to the scale 1:25000, describing in two separate sections the generalization of ordinary roads and the generalization of highways.

One of the main aspects of the process is how to select the roads to generalize and those to delete. In the following sections it will be explained how we could drive the selection process by enriching the input data model on the base of morphological analysis of the roads.

6.2.1 Related work

Morphology, intended as the study of shape and form, is an important topic in the field of generalization: one of the main aims when generalizing a map is in fact to maintain the form and shapes represented in the input map.

The shape and form of the features have been studied, measured and characterized (see for example [Agent 2000]) and researches have been done to understand the perception of shapes and forms [Wertheimer 1938], [Thomson and Richardson 1999], [Thomson and Brooks 2000]. All these information have been usually used to drive the generalization process; in the examples that we will present in this chapter instead, we use morphology at an earlier stage, to reclassify features or to refine the existing classification in order to gather a better knowledge of what is represented on the map and operate a more conscious generalization.

The research on the analysis and generalization of road network has been very intense too, because of the main role played by roads in maps. Most of the authors working on road networks use the concept of strokes derived from the work of [Thomson and Richardson 1999] on perceptual grouping: in a road network represented by a graph, we can define a stroke as a chain of edges that are joined on the principle of good continuation. The concept of strokes is much used both in the analysis and in the generalization of road networks; strokes are usually built on the basis of straightness, but in some cases other data can be taken in consideration, using information directly from the input model (e.g. road names), or enriching the data calculating new metrics [Claramunt 2004], [Heinzle 2005].

Although strokes are a very important tool in generalization, strokes alone can not provide a complete solution to the generalization of road junctions and highways; some works that address more specifically these topics are those of [Mackaness and Mackechnie 1999, Thom 2005, Touya 2007].

In [Mackaness and Mackechnie 1999] the authors propose an interesting approach to generalize road junctions, using cluster analysis and graph theory. Their idea is to find the road junctions as the regions where the nodes of the road network are denser. This is done clustering the vertices of the road network and applying a "granularity" threshold to create the clusters. Every cluster represents a road junction. The graph representing each junction is then created and contracted: the junction is simplified collapsing all the vertices of the cluster to the centroid and connecting all the edges to it. Changing the granularity threshold is possible to control the level of generalization of the junction, by collapsing more or less vertices. Although the algorithm proposed to detect and generalize road junctions produces viable results, the choice of the right granularity is still an open question; furthermore, as noted by the same authors, in some instances the results were not acceptable leading to what they defined "the collapsing star effect".

In [Thom 2005], the problem of collapsing dual-carriageway is addressed with a four-steps algorithm that builds the strokes from the road sections, pairs the strokes, collapses each pair and connects the resulting line work with the remaining road network. The author notes that because the direction of the slip roads is almost tangential to the main roads, building the strokes only on the basis of straightness leads to unpredictable results. This problem is solved using the direction of the road (stored in the input data model) to develop a method of tracking one-way sections.

In [Touya 2005] the author describes a full and generic process to allow road network selection in model generalization. The author orchestrates many different algorithms in a process entailing four steps: data enrichment through structures and pattern recognition, rural selection based on assessing traffic by shortest path computing, street selection algorithm based on road block aggregation and structures typification. The classification of road junctions is achieved by classifying first simple road junctions analyzing, at every node of the road graph, the angles between the incident edges; complex road junctions can be then found as particular aggregation of simple ones. Unfortunately this classification process is not explained in details.

6.2.2 Generalization of ordinary roads[5]

The process of road generalization deals usually with two main aspects: road selection and displacement. Road selection allows reducing the complexity of the road network preserving a smaller set of roads; displacement instead allows to solve the dispute for space due to the symbolization of the map objects; the latter topic, however, goes beyond the scope of our research.

Working at large-medium scales the problem of selection is simpler, as many of the roads in the source are retained at the target scale; nevertheless the large scale of the source data brings some problems: road junctions are represented with too many details, that need to be “filtered” at the target scale; for this very purpose a road junction generalization algorithm has been developed.

The generalization of the ordinary roads comprises then four main steps:

simplification,
harmonization,
removal of dangling edges,
generalization of road junctions.

6.2.2.1 Simplification

The algorithm developed to simplify the source data applies the Douglas-Peucker (DP) algorithm to the edges of the graph representing the road network. Although the threshold used is small and corresponds to the accuracy of the target data model (2.5m), in the cases of narrow roads it might be big enough to make the road intersect a neighboring object (e.g. a building). To avoid this event, the algorithm bounds the allowed shape of the generalized line to the interior of the polygon that represents that section of road in the GeoDBR: in the case the generalized line intersects the boundary of the polygon, the line is iteratively returned to its original shape and simplified applying a smaller threshold to the DP algorithm.

[5] This work was done with Igor Lissandron, University of Padua (see [Savino et al., 2009]).

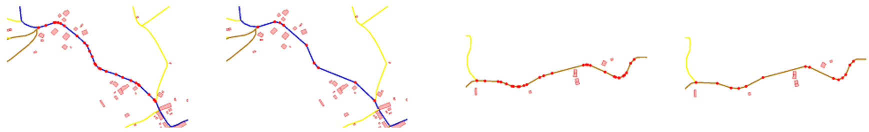


Figure 16: applying Douglas-Peucker to roads

Figure 17: example of road harmonization: the edges in red (left) have been classified as those adjacent (right)

6.2.2.2 Harmonization

The harmonization algorithm extends the classification of a road to its neighboring roads, with the purpose to have a more uniform classification on contiguous roads.

In the source data, roads are represented by the edges of a graph; each edge has its own classification and it may happen that two adjacent edges belongs to two different road classes. The idea behind the algorithms is that the class of a road should be constant along all the edges that compose the road and that if a class change should happen, it should take place anywhere only in presence of a special condition (i.e. the intersection with another feature class). This idea comes both from common sense and from the opinion that generalization should reduce details [Mackanness, 2008], like an excessive segmentation of roads.

Furthermore the analysis of the source data highlighted the presence of errors in the classification of roads as sudden changes in the road class; since the road class is used for the construction of the strokes [Thomson and Richardson, 1999], correcting these errors through harmonization will also improve the results of generalization.

The idea has been translated in a simple algorithm that works on the strokes built on the edges of the graph: for each stroke that is adjacent, on both sides, to two strokes having the same road class, but different from its own, the stroke is harmonized, i.e. its road class is changed to that of the adjacent strokes.

The harmonization is limited by two conditions:

1. the stroke to be harmonized should be shorter than a threshold (1500 meters)

2. the stroke should be the "good continuation" of the adjacent strokes (i.e. if their road class was equal, the stroke and the two adjacent strokes, one on each side, would have been part of one single stroke)

Harmonization is also applied to dangling strokes: in this case the same conditions apply but there is only one adjacent stroke; the harmonization will change the road class of the shorter of the two.

6.2.2.3 Removal of dangling edges

According to the IGMI specifications, roads shorter than 250 meters should not be generalized to the DB25. This rule is applied only to dangling edges of the graph, as it would cause the loss of connections in the network if applied on all the edges. As the removal of a dangling edge may create a new dangling edge, the rule is applied recursively; the recursion ends when the length of the edge to delete added to the lengths of all the adjacent edges already deleted is bigger than the threshold of 250 meters (this to avoid that a dangling sequence of edges, each shorter than 250 m, would be completely deleted by the recursion).

According to the IGMI specifications, the removal of a dangling edge is subject also to another condition: it should not be deleted if it is the only access road to a building or group of buildings.

To comply to this requirement, for each dangling edge candidate to deletion the algorithm follows these steps:

1. a buffer of size R is drawn around the candidate dangling edge to search for any building closer than R to the road
2. if such a building is found, a buffer is drawn around the building, to check whether another road passes nearby the building
3. if such a road is found and it is not dangling, the candidate edge is deleted; if the second road is another dangling edge, only the longest among the two is kept.

The algorithm actually performs this control not only on buildings, but also on groups of buildings (the grouping of building is similar to that described in [Boffet, 2001]: a buffer is drawn around each building and overlapping buffers are merged: the single buffers identify single buildings, while merged buffer identify clusters of buildings).

Figure 18: removing dangling edges. Left: in red dangling edges candidate to deletion (after the first iteration). Right: final result after the iterative procedure: all the dangling edges not leading to a group of buildings or being a redundant connection to them have been deleted

6.2.2.4 Generalization of road junctions

The IGMI specifications for the DB25 states specifically that every roundabout of radius smaller than 25 m should be collapsed to its centroid and, more generally, that road junction should be "simplified". To fulfill these specifications it is necessary to be able to recognize both the roundabouts and the "complex" road junctions (i.e. those needing to be simplified): as the source data model does not provide such information, it was necessary to develop an algorithm able to detect these structures in the road network.

When generalizing road junctions, the first problem to solve was to detect those that needed to be generalized. This was like asking: "what makes some road junctions so complex that they need to be simplified?" The answer that we found is "redundancy": the difference between a "simple" junction and a "complex" one is the presence of short edges (e.g. slip roads, access ramps) that create redundant connections in the graph.

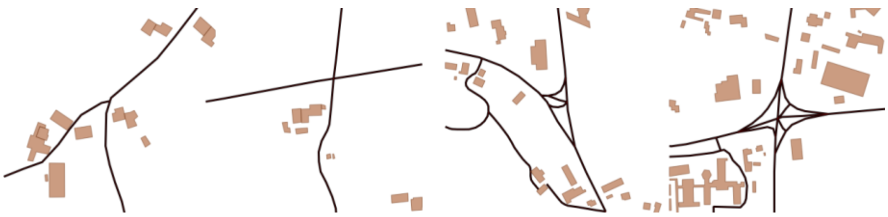


Figure 19: simple (two leftmost) and complex (two rightmost) road junctions

Since a redundant edge in a graph create a cycle, our algorithm finds the junctions to generalize by looking for all the cycles in the road graph; of course, as the road graph is highly cyclic, we had to set a threshold: we empirically set it as 250 m of maximum perimeter length. What the algorithm finds is a set of cycles of different sizes and shapes that may be isolated or adjacent to other cycles.

The most recognizable junctions are probably the roundabouts: testing the "roundness" of every cycle (perimeter to area ratio similar to 4π/p) we could easily find them; this however left many cycles still unclassified.

As it was clear by visual inspection of the results, some of the cycles found were part of more complex junctions. Then, in order to look at the broader picture, we merged together all the adjacent cycles and calculated how many points the boundary of the resulting merged cycle had in common with the road graph: we found out that the number of these points (called special nodes) and the type of junction represented by the merged cycles were related and so this could be a good way to classify them.

We built the strokes on the basis of the gestalt principle of “good continuation” connecting the most straight chain of edges passing through the special nodes. Strokes could be built just “locally” on the road edges touching the road junction; from our experiments, the best results were obtained not considering any semantic information (e.g. road name or classification) of the edges: in some cases, in fact, the original classification changed right after the road junction, thus preventing the construction of longer strokes.

Depending on how many special nodes a stroke was crossing (one or two), we classified the strokes as crossing roads (crossing 2 special nodes) and incoming roads (crossing only 1 special node). All the remaining strokes were classified as internal roads.

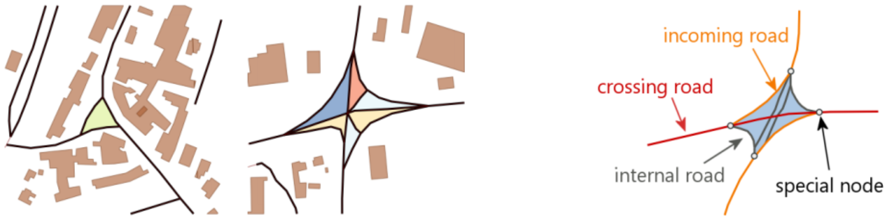
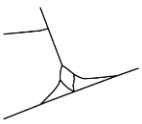


Figure 20: left: road junctions are detected using road cycles; each road junction can be formed by one or more road cycles; right: special nodes, incoming roads, crossing roads and internal roads in a road junction

On the basis of the number of special nodes and the number and type of strokes of each junction, we could further classify the road junctions in:

T-junctions
Paired T-junctions
Crossroads

Each road junction not falling in one of these classes is tagged as “unclassified junction”.

	Junction type	Number of special nodes	Number of crossing roads	Number of incoming roads
	T-Junction	3	1	1

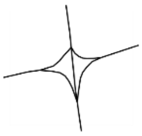
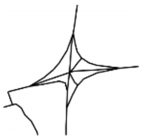
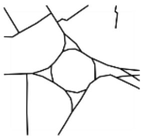
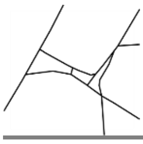
	Paired T-Junction	4	1	2
	Crossroads	4	2	0
	Roundabout	(classified at earlier stage)		
	Unclassified	(any junction not falling in the criteria above)		

Table 1: relation among the type and the number of special nodes, crossing roads and incoming roads for each type of road junction

Visually inspecting the results of the algorithm, we found that it performs in accordance with the expectations: roundabouts, T-junctions, paired T-junctions and crossroads are correctly detected and classified most of the time and what the algorithm tags as “unclassified junction” are usually junctions that are arguably difficult to classify, even for a human. In some cases, though, there are some false negatives: T-junctions, paired T-junctions and crossroads can end up in the “unclassified” group because of a single edge touching the boundary of the merged cycle, thus increasing the number of special nodes over the thresholds. False positives can also happen, in particular road cycles with three special points can be mistakenly classified as T-junctions. A concavity test is used to avoid this case: since a real T-junction should have slip roads to connect smoothly the crossing road with the incoming road, and slip roads by design have a concave shape, the merged cycle of a real T-junction should be contained by a triangle drawn on its three special nodes. Empiric tests revealed that it is sufficient to compare the area size of the triangle built on the three special nodes with that of the merged cycle to filter out false T-junctions.

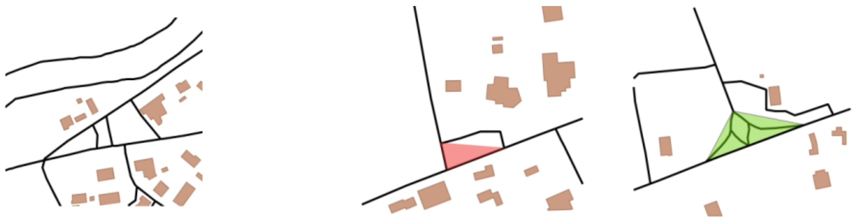


Figure 21: (left) a T-junction is treated as an "unclassified junction" because the edge indicated with an arrow increases the number of special nodes. (right) testing two candidate T-junctions: the first fails the concavity test (red triangle), the second passes (green triangle)

At the end of the process, all the road junctions have been classified into 5 categories:

roundabout
crossroad
T-junction
paired T-junction
un-classified junction

For each of these categories a specific generalization algorithm is executed. The main idea is to remove all the roads that are not relevant in the junction, although preserving the functionality of the junction: this is achieved carrying out a test prior to the elimination of every segment to verify that its removal will not lead to a lack of connectivity among the distinctive nodes of the junction. As mentioned before, all the generalization algorithms operate just only inside the perimeters of the joined loops, assuring that no topology changes are made to features lying out-side these boundaries.

Roundabouts

Generalization of roundabouts differs depending on their size and on the presence of road loops around them. The size of a roundabout is calculated as a "virtual radius" R, that is its perimeter divided by 2π. Following DB25 specifications if the radius is smaller than 25 meters the roundabout is collapsed to its centroid, otherwise it is replaced by a perfect circle with the centre in the centroid and radius R. Any road loop touching a roundabout is also generalized: from each distinctive node not touching the roundabout a line is drawn to the road loop centroid and from here to the roundabout centroid. If the road loop is part of a joined loop, the line from the road loop centroid is connected to the joined loop centroid and then to the roundabout centroid.

In case the roundabout has not been collapsed, the lines to its centroid are cut on the circumference of the roundabout. This same procedure is applied, of course, also for any road merging into the roundabout.

T-junctions

Regarding T-junctions, there are two different generalization procedures.

T-junctions are junctions where a road is connected with two or more access lanes to a crossing road. According to the definition, also road loops made by one road having both the end points connected to two contiguous roads or to the same one are classified as T-junctions: this kind of road loop, that we call “redundant loop” actually doesn't represent a real junction and should be processed in a different way. To distinguish between redundant loops and T-junctions a morphological test is executed: in a real T-junction the incoming street smoothly merge into the crossing street through some access lanes (they need to be at least two in order to build a loop and be detected), thus giving the joined loop a concave shape; redundant loops, on the contrary, often shape a convex joined loop. Figure 7 (a) and (c) clarifies this concept.

On the basis of this consideration we can distinguish between real T-junction and redundant loops by a simple concavity/convexity test. This test is done by comparing the joined loop area to that of the polygon built using the distinctive nodes as vertices: if the latter is bigger then the joined loop it is concave and we have a real T-junction, otherwise we have a redundant loop. Regarding T-junctions, the generalization is achieved through the preservation of the crossing road and the removal of all the other segments generating the joined loop; one median confluence segment is created between the two most external access ramps. Redundant loops are solved by simply deleting the internal road and thus “opening” the loop.

Paired T-junctions

Paired T-junctions are processed through the removal of all the access ramps and the creation of two segments starting from the distinctive nodes of the two incoming roads and ending in a common point over the crossing street, this point being the centroid of the joined loop. The choice to manage this kind of junction as a class instead of simply managing it as two singular T-junctions was taken to avoid the creation of two distinct intersections on the crossing road that, depending on the direction of the confluence segments of the two singular T-junction, could be too close to each other.

Crossroads

Crossroads are the simplest class to solve: the two crossing roads are preserved while all the other internal roads in the joined loop are removed; this operation corresponds to removing all the access ramps and confluence lanes of the junction, leaving only the main roads.

Un-classified junctions

Regarding un-classified junctions, a best effort generalization procedure is applied: the algorithm removes all the internal roads of the joined loop, thus deleting some of the loops and simplifying the overall geometry.

6.2.3 Generalization of highways[6]

In Italy highways are a special part of the road network: they run isolated from the ordinary roads and the connection to them must pass through a toll gate; the highway network can be considered then a sub-graph of the whole road network. The most relevant features in the highway network are the two carriageways: other features are connected to them, as rest areas, slip roads and toll plazas. In our input data model, all the edges belonging to the highway graph are only classified as "highway" and not further specialized.

The generalization of highways was hindered by a problem between the source and target data models: in the IGMI DB25, there exists a specific object for the highway toll stations, the highway slip roads, the highway rest areas and the highway carriageways whereas the GeoDBR lists only a feature "highway", not further specialized, from which to derive all those objects.

This problem has been successfully solved developing a data enrichment process relying on the study of form and shape of the edges composing the highway graph.

The first step of the process is to find the main carriageways of the highway; following the slip roads are found, leaving rest areas and toll stations to be classified last.

6.2.3.1 Classification

The first thought one has when thinking of a highway is something long, continuous and straight; this remark let us to move our first step toward the solution: we found among all the edges the longest and the most straight and we classified it as "carriageway". This first edge was used as a "seeding edge": starting from it we grew the carriageway adding all the edges connected to it first in one direction and then in the opposite. This procedure went on until a fork was met.

A fork in the highway means either that there is a slip road joining or leaving the carriageway, either because the highway splits in two directions or because there is an exit. As noted by others [Thom 2005], because slip roads are by design close to tangential when joining or leaving their dual carriageway, straightness alone is not sufficient to create reliably strokes from dual carriageways. Because of the function of slip roads anyway, their property of being tangential is required

[6] This work was done with Matteo Zanon, University of Padua (see [Savino et al., 2010]).

only locally, in close proximity to the junction with the carriageway: looking “further away”, the slip road changes its direction (e.g. to route the traffic to a rest area, a toll station, or another highway).

To construct the strokes from the carriageway we devised a metric, called *bend ratio*, that takes into account the way the direction of an edge varies. The bend ratio of an edge A composed by n vertices a_0, a_1, a_2, …. a_n is defined as

$$\text{bend ratio} = \frac{\sum_i L_i \, diff_i}{L}$$

where:

L is the length of the edge A
L_i is the distance between two consecutive vertices a_{i-1} and a_i
$diff_i$ is the difference between the angle of the segment from a_{i-1} and a_i and the angle of the segment from the first to the last vertices of the edge A

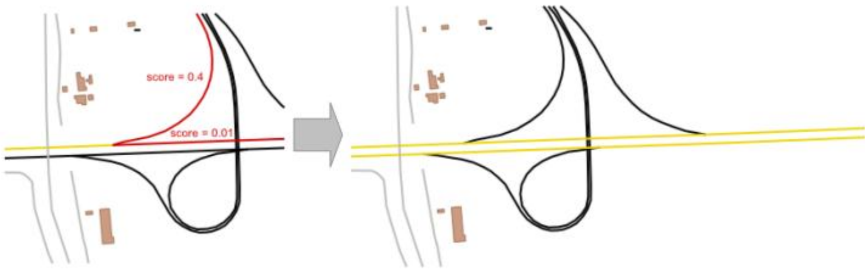


Figure 22: the score of the bend ratio is used to build the strokes from the carriageway

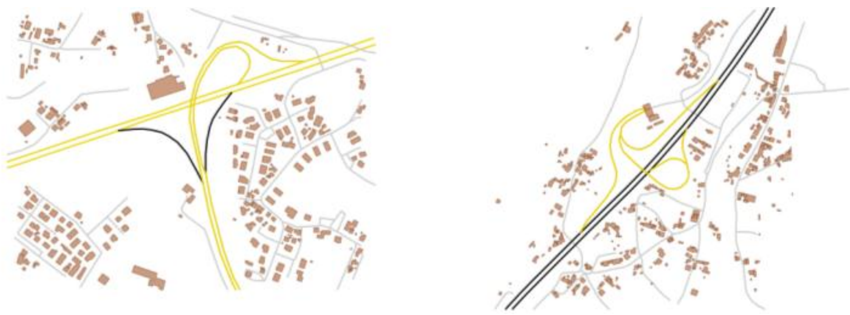


Figure 23: classification of carriageways (left) and slip roads (right). In the pictures, in gray the network of ordinary roads, in black the highway network, in yellow a group of carriageway (left) and some slip roads (right)

The value of the bend ratio increases the less the edge is rectilinear.

Using the bend ratio slip roads can be distinguished and the construction of the stroke continues along the carriageway; when a carriageway cannot be extended further, a new seed edge is searched among the edges not tagged and the process starts again.

At the end of the process all the carriageway and the slip roads connected to them are identified and classified, but there are still other objects to be recognized: rest areas and toll stations. Some rules derived from conditions that apply to them helped us in doing so:

toll stations are the only edges of the highway graph allowed to be connected with the ordinary roads and are either at the end of a carriageway or connected to a slip road;

rest areas are portions of the highway graph that are connected only to the same carriageway and can not be connected to normal roads

These restrictions apply in Italy, but other more general characteristic can be used to classify them: for example in a toll plaza the presence of the toll booths divides the carriageway in many lanes, while a rest area can be identified by the presence of a graph that has sharper bends -since it's intended for low speed traffic.

Applying the conditions above, also toll plazas and rest areas have been classified.

Having set up a set of rules to identify the objects we are interested in, we can then be confident that one edge not matched by any of these does not need further attention and can be discarded: this is an important aspect to consider as it helps to avoid that errors in the input dataset are passed to the generalized one (see Figure 25 left).

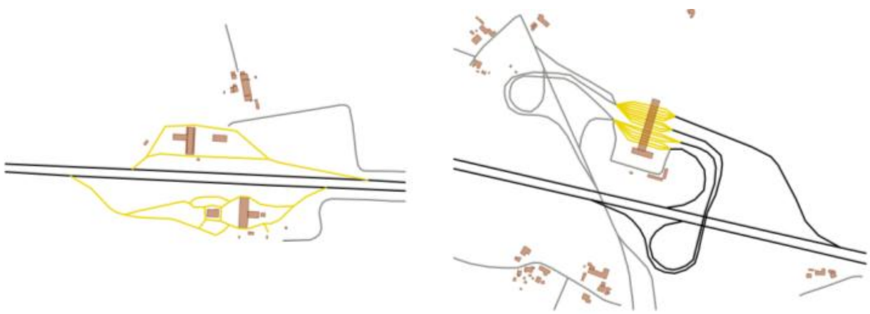


Figure 24: rest areas (left) and toll stations (right) can be recognized due to their particular shape and their connection with carriageway and the road network. In the pictures, in gray the network of ordinary roads, in black the highway network, in yellow a rest area (left) and a toll station (right)

6.2.3.2 Generalization

At the end of the classification process, the highway has been decomposed in:

carriageways,
slip roads,
toll-plazas,
rest areas.

Each of these elements is generalized with a specific algorithm; the results of this process are shown in chapter 7.

Carriageways

Complying with the DB25 data model, dual carriageways should be collapsed to a single line running in between them, unless they are separated by a distance bigger than a certain threshold (30 meters). The algorithm performing this operation finds the sections of the highway where the two carriageways are closer than this threshold and draws a middle line between them; the two sections will be deleted and the middle line will be connected with the rest of the highway graph. In connecting the generalized carriageway to the highway graph particular care is taken to draw a smooth transition between the former and the latter. In the case the two carriageways run at a distance from each other close to the threshold, it might happen that the generalization results in a sequence of collapsed and not collapsed sections ("sausage effect"); to avoid this event a minimum length constraint is enforced on the not collapsed sections: a sequence of collapsed and not collapsed carriageways would be then represented only if the not collapsed section is longer than 1 km (this value was found empirically).

Figure 25: (left) an error in the classification of the input data is detected: the edge classified as highway (red) is too short to be a highway and, as a slip road, it is not connected to anything; it will then not considered part of the highway. (right) The classification of slip roads and carriageways is reliable also in complex cases.

Toll-plazas

In the source data, nearby the toll-gates the highway graph usually branches in many edges (see Figure 24 right). These many edges are then generalized, reducing their number by merging together and finally collapsing to a line the road cycles created by the graph branching around the toll gates. This procedure might be thought as a sort of typification, as it reduces the number of objects while maintaining the pattern.

Rest areas

According to the specifications, in the DB25 rest areas should be represented as polygons or its centroid in the case the polygon is smaller than a threshold (3000 sqm). The classification process identifies the rest areas as sections of the highway graph, and thus as lines. The generalization process then has to create the polygon of the rest area starting from these lines: this is easily accomplished by drawing the road cycle formed by these lines and the carriageway; the size of the polygon built using the road cycle is then measured and if smaller than the threshold, the polygon is collapsed to a point.

Slip roads

The generalization of the slip roads is similar to that of the carriageways: sections of the slip roads that are to close are collapsed to their middle line and a smooth connection between the slip roads and the carriageway is drawn after the latter or the former has been collapsed to a single line.

6.3 Generalization of the hydrography network7

With no surprise hydrography is one of the most important feature classes to generalize. Hydrography as a theme comprises many feature classes that describe both man made and natural features, flowing and still waters, with the network of flowing waters usually represented as a node-edge graph. As one of the most prominent natural themes, hydrography is probably the most complex, for its big number of elements and feature classes, its extension spanning big areas of the dataset and its being often in relation with other themes (as transport networks and settlements). The generalization of hydrography is then a complex task, which made it one of the first topics to focus our attention to.

6.3.1 Input and output data model

The GeoDBR and DB25 data models for hydrography share many similarities: they both distinct watercourses between man-made (canals) and natural (rivers), and classify them on their width. All the flowing waters are represented in a node-edge graph and the broadest of them (both canals and rivers) are also represented as an area.

---

[7] This work was done with Giovanni Langiù and Fabio Canton, University of Padua (see [Savino et al., 2011b]).

In the input model there is a 1:1 relation between edges and areas of the broadest watercourses (the area of a broad watercourse is divided into sections, each of them containing one edge of the graph). A node is present at each intersection between two or more edges and also at the intersection with the edges of other graphs, noticeably those of road and railroad networks.

Water bodies like ponds, lakes, swamps and so on are represented in separate feature classes and those of them connected to the hydrography network contain also edges that guarantee their connection to the graph (e.g. edges connecting the inlet and outlet of a lake). Smaller watercourses, as creeks and ditches, are stored in two different feature classes; creeks are part of the hydrography graph while ditches are not.

Both the input and output data models for the feature classes representing flowing waters (rivers and canals) have a simple semantic, comprising only the attributes id, name and hydrography class (a code to differ between the types of rivers or canals based on their width).

The flow direction is not present in the data models and this will require reconstructing it during the cartographic generalization process.

The differences between the two data models arise when comparing the specifications of each feature class. In the input data model (GeoDBR) watercourses are classified into only 2 hydrography class: "narrow" watercourses, represented as a single line, and "broad" watercourses, those wider than 1 meter, represented both with a single line and an area. In the output data model (DB25), they are divided in four hydrography classes: "very small", "small", "medium" and "large" watercourses, respectively with width being less than 1 meter, from 1 to 5 meters, from 5 to 20 meters, more than 20 meters; all of them are represented with a line and only the latter class is represented also by an area. As a further specification, the IGMI imposes a minimum river and canal length of 250 meters.

6.3.2 Objectives

Comparing the input and output data models and their specifications, it was clear that the generalization process for the hydrography should pursue two objectives:

to find a way to classify the watercourse on their width;
to apply the 250 meters length threshold to each watercourse, i.e. to prune the network of those branches shorter than this measure.

Furthermore, a third objective was set by the IGMI specifications, requiring that:

in the regions where the hydrography network was too dense, the less important branches should be pruned.

In the following sections we will describe the generalization process of the river network. Water bodies as lakes, ponds, swamps will not be covered. Even though both the model and cartographic generalization process could be applied to canals, we will not make explicit reference to them. The generalization of ditches will be treated separately.

6.3.3 Related work

When we started to develop our process for the hydrography generalization we looked with interest at the experiences done in the past by others. The objective that we pursued was three-folded: first we had to find a way to classify the rivers on their width, second prune the network of the shortest branches and third prune the network of the least important branches in regions where there were too many branches. A great deal of work on generalization of hydrography can be found in literature, especially on cartographic generalization.

[Horton, 1945 and Strahler, 1952] developed a metric to classify the branches of a river network using a counter that increases when two branches meet. This metric, known as Strahler order, is widely used to enrich the river data models extending the original classification to prune the network [Thomson and Brooks, 2000, Touya, 2007]. To prune the river network many other parameters and thresholds can be used, singularly or in conjunction: density [Stanislawski, 2008], water basin [Tinghua & al, 2006], the upstream drainage area [Stanislawski, 2009] or other values [Zhang, 2007], [Brewer & al. 2009].

As a matter of fact most of the river network generalization algorithms step first through a process of data enrichment and then prune the river network. In the following section we will illustrate the generalization process that we developed: our set up follows some of the ideas and concepts found in literature but also introduces some original ideas.

6.3.4 Width measurement

The width of a river, although being a concept easy to understand, is an ambiguous definition when it has to be transformed in a metric for an algorithm. The width of a river, in fact, is not represented as a single value, but it varies along the course of the whole river. Since in our datasets each river is divided in sections, each of them comprising both an area and an edge of the graph passing through it, we decided to measure the width of each section. This task can be accomplished in many ways, more or less precise (e.g. with a perimeter and area ratio): our choice was to develop a simple algorithm, mimicking the manual process of river width measuring. The algorithm to measure the width of each river section area samples the edge in n points equally spaced d meters. For each of these points a line normal to the direction of the edge in the point is drawn and the first couple of intersection points with the boundary of the area are found; the width of the section will be the average of the distances between each couple of intersection points.

Once the width of each section has been calculated, each section can be classified according to the target data model, following the IGMI specifications; if due to model generalization the geometry of a river section has to be collapsed from area to line, the algorithm simply deletes the area and keeps the relative edge.

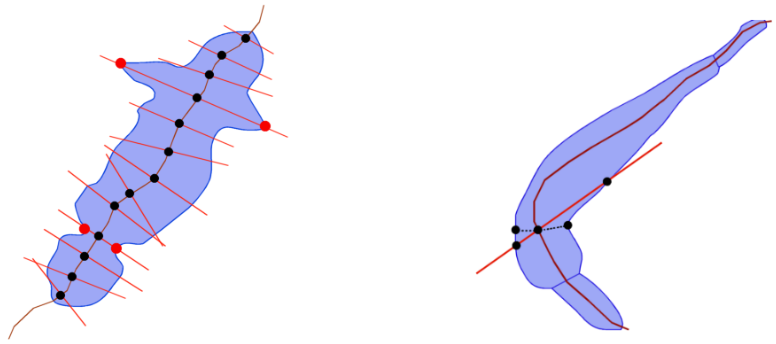


Figure 26: on the left: points of width local minima and maxima (red); on the right: the normal projected from a point i on an edge that has many bends can return a width measure that is much different from the distance between i and the two closest points on the boundary of the section area

The choice of averaging the width measures in each section could be questioned: indeed the search for a minimum or maximum value could be a valid alternative too and maybe provide some more insightful information (e.g. the minimum width is related to the maximum allowed boat size).

Anyway our guess was that the average would be a more reliable measure, being robust to eventual local minima or maxima in which our simple algorithm could fall (see Figure 26 left); furthermore since each river is divided into a new section at every confluence, the size of each section will be more or less constant (the water flowing in or out of if is the same), with the average width then being not too far away from the real minimum and maximum width.

Moreover some errors in our measurement could be tolerated, as our aim was to calculate a reference measure to classify the rivers and not to calculate their exact width.

The simple approach that we set up gives the best approximation of the width of each river section when the shape of its area is mostly regular -i.e. when the river banks run parallel to the edge-. Results can be worse for odd shaped sections or sections that are small compared to the parameter *d*. For the former of these issues our guess was that averaging the width measured at each of the *n* points will mitigate these measurement errors; to solve the latter we set up a harmonization process that will be described next.

6.3.5 Harmonization

After the classification process ended, we realized that we needed a harmonization algorithm to improve the quality of the result as in some cases the class change of a river section lead to aesthetically unpleasant results. The problems that we found have to different reasons:

1. area to line collapse of small sections of braided river
2. rough appearance of the confluence of an area river section into a line river section

In both cases the problem was that the area to line collapse of a river section due to the new classification caused a sudden change in the representation of the river (see Figure 27); the solution to such problem was to analyze the neighbors of each river section that had been collapsed and in some cases to override the new classification or to change the shape of the neighboring areas.

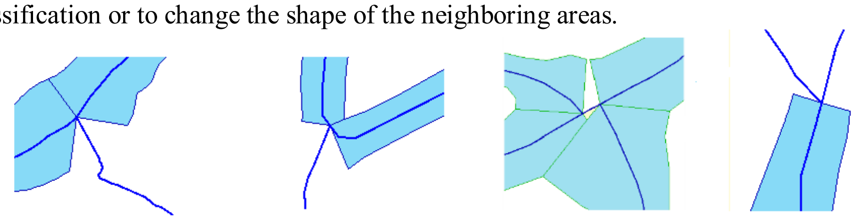


Figure 27: river sections requiring harmonization

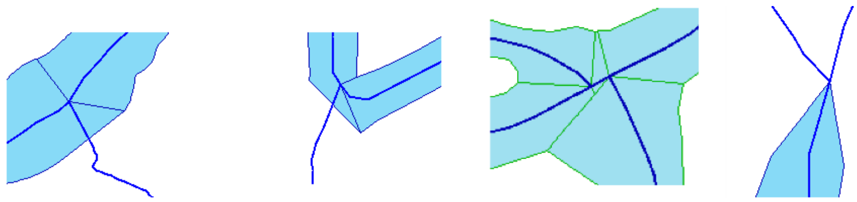


Figure 28: river sections after the harmonization process

In braided rivers the watercourses are segmented in many sections due to the big number of edge intersections and this resulted in sections too small to be correctly classified: it might happen then that a small section connecting two confluences could be classified as a minor river[8], and then collapsed to a line, creating an abrupt change in the shape of the river. (see Figure 27, second from right) In this case the classification of this small section has been overridden and the section "upgraded" to "big" river, returning it to its original representation of area river section.

In other cases, the confluence of a section that has been collapsed to line with other area river section could too result in a rough change in the representation. This is the case of a minor river leaving or entering a "big" river or a minor river

[8] We use the term minor river for rivers classified as "very small", "small" or "medium"; i.e. represented only by a line and not also by an area.

becoming a “big” river. In such cases the harmonization algorithm will change the shape of the area river sections touching the minor river in order to assure a smoother representation of the class change (see Figure 28).

6.3.6 Pruning

As expected, pruning the river network was a challenging task.

The IGMI specifications for the DB25 rivers require that watercourses shorter than 250 meters should not be acquired. This threshold can not be applied directly to the data because deleting all the edges shorter than 250 meters would disconnect the graph of the river network. This constraint could neither be translated in deleting all the dangling edges shorter than 250 meters, as this will much probably cause all the graph to shrink.

The IGMI specifications also require to prune the network of the least important branches in dense regions. This implies the ability to recognize the importance of each river and compare it. The four different classes of the IGMI data model are not enough for such a task: a new classification is needed, that could take in account other parameters to evaluate the importance of each river.

The solution was to set up a network pruning algorithm able to reconstruct the course of each river, to enrich the data in order to classify the rivers by importance and to prune the network, first applying the length threshold and after removing the least important branches in the densest regions of it.

All these considerations lead us to develop a pruning algorithm that follows these steps:

Reconstruction of flow direction,
Data enrichment,
River course reconstruction,
Short rivers pruning,
Dense rivers pruning.

6.3.6.1 Reconstruction of flow direction

The first step to reconstruct the course of a river from its sections is to know the direction that water is flowing to. Without this piece of information it is impossible to say, at a branch, if two sections are converging in one or one is branching in two. Flow direction is usually embedded in the hydrography data models, either implicitly (e.g. the flow direction follows the order of the points of each edge) or explicitly (i.e. as an attribute). In the case the flow direction is not known, it can be calculated from the z-coordinate. This was the case with our input data.

The algorithm to evaluate the flow direction in each section of the river network iteratively analyzed every edge and for every current edge tried to divide the edges touching it into fathers, children and siblings, comparing the z-coordinate of the vertices of each edge.

For the current edge the algorithm finds the highest point *cpMax* and the lowest point *cpMin*; the same is done for all the edges connected to it:

if an edge has pMax > cpMax and it is connected to c on the point cpMax, it is a father
if an edge has pMin < cpMin and it is connected to c on the point cpMin, it is a child
if an edge has pMax > cpMin and it is connected to c on the point cpMin, it is a sibling
if an edge has pMin < cpMax and it is connected to c on the point cpMax, it is a sibling

Furthermore

if c doesn't have any father, it is a source
if c doesn't have any children, it is a drain
the flow direction of c is from cpMax to cpMin

It is common that in cartographic datasets the values on the *z* plane have a lower precision than the *x, y* data. For this reason the simple model above had to be expanded to include two special cases:

flat edges
uphill edges

A flat edge happens when *cpMax* = *cpMin*. This may be caused when a river flows on an almost flat surface (e.g. a big plain) or the edge is an artificial connector inside a water body (e.g. in a lake), or the edge is too short to record a difference in the *z* values of its points. Flat edges are quite common in braided rivers where the slope is not very steep and the length of the sections is small.

Uphill edges happens when an error in the z values of an edge turn its flow direction uphill. This error, caused by a high tolerance on the z values precision, is very difficult to identify. In an uphill edge, *cpMax* and *cpMin* are inverted and fathers and children edges are classified as siblings, actually disconnecting the graph and preventing the generalization algorithm to work on all the edges down hill from the current edge.

The algorithm tries to detect the uphill edges: if an edge has only siblings, the connected edges are classified again inverting *cpMax* and *cpMin* and if this new classification provides at least one father and one child, the edge is flagged as uphill and the latter classification is kept.

Unfortunately not all the uphill edges can be detected; since flat edges are much easier to treat, it has been chosen to reduce the number of possible uphill edges using a z threshold *zT*: if *cpMax* - *cpMin* < *zT* then the edge is classified as flat. This results in many edges, both uphill and correct, being treated as flat: we somehow traded many good edges and some bad ones for plenty of flat edges, but this is definitely worthy, as flat edges will not usually block the generalization process.

Even though it is not possible to set the flow direction on a flat edge, it is still possible to classify the connected edges following the first six rules above. One more rule is used to deal with flat edges:

if c is connected to a flat edge d, the edge connected to d should be considered connected also to c

This rule virtually collapses each flat edge to a point, connecting together its fathers and children. In a group of connected flat edges like those inside a lake, this rule means that the inlets of the lake are directly connected to its outlet.

With the eight rules listed above and no uphill edges above the z threshold *zT* it is possible to find in the river network sources, drains and find, for each edge, its fathers and children.

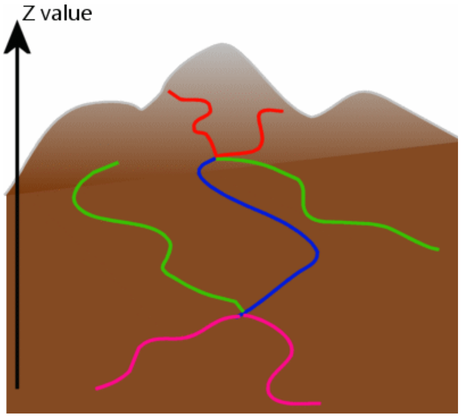


Figure 29: fathers (red), children (purple) and siblings (green), of a river (blue)

6.3.6.2 Data enrichment

The data enrichment process collects, for each current edge *i*, the following information:

S_i : the Strahler order of the edge *i*
L_i : the total distance to the furthest source up hill
B_i : the total number of branches up hill

The procedure is top-down: starting from one of the sources, the algorithm follows the flow direction down hill, calculating Li for each current edge (S_i and B_i will not change until another edge is met). If the current edge c has two fathers a,b, the values for c will be calculated as follows:

Sc = max(Sa, Sb) *if Sa != Sb*
Sc = Sa +1 *otherwise*

$L_i = max\ (\ La,Lb\ ) + length(\ i\ )$
$B_i = max\ (\ Ba,Bb\ ) + 1$

The next current edge will be one of the children of c. The algorithm randomly chooses a source edge to start from: as a consequence it may happen that when two edges meet, one of them has not been enriched yet. In such a case the algorithm will pick another source and start the procedure from there. The same happens if the current edge doesn't have any child. The process will end when all the edges have been enriched.

6.3.6.3 River course reconstruction

The procedure is bottom-up: starting from the drain with the highest Strahler order, the algorithm follows the flow direction up hill choosing as the next current edge one of the fathers edge of the current edge. The next current edge is chosen after assigning a score to each father that depends on the name of the edge, the IGMI hydrography class it belongs to (among "very small", "small", "medium" and "big" river), the number *L*, the number *B* and its width.

The scoring procedure will assign a higher score to a father edge if:

has the same name of the current edge
belongs to the highest IGMI hydrography class
has the highest value *L*
has the highest value *B*
has the largest width

The score will be decreased if the father edge

has a different name from the current edge
has a lower Strahler order than the current edge

The ratio behind the scoring is to try to find, at each fork, the most relevant branch of the river. The scoring mechanism of course is not perfect but our aim is not to reconstruct perfectly the course of a river, but to have a good approximation.

If the current edge has not any father edge, the course of this river has been reconstructed to its source: all the edges touched have been flagged and added with an unique id; the procedure can start again, selecting from the edges not yet flagged that with the highest Strahler order.

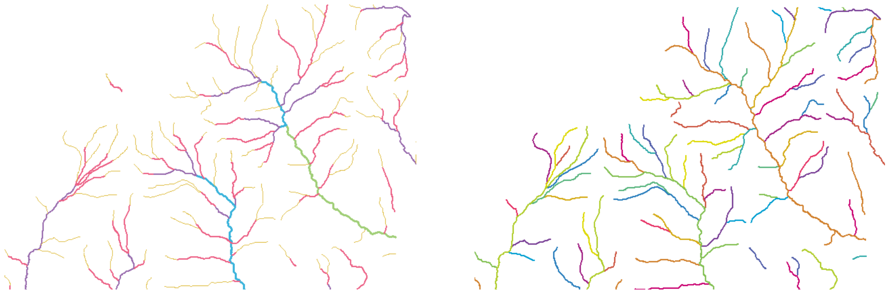


Figure 30: left: Strahler order: different color mean different order; right: reconstructed river courses: different color means different river course

6.3.6.4 Short rivers pruning

Once that the course of each river has been reconstructed, it is straight forward to apply the IGMI minimum length threshold: the length of every river course is calculated as the sum of all its sections and if the length is smaller than the threshold, all the sections belonging to it are deleted. Recursively, every river stemming from a river that has been deleted is deleted too.

6.3.6.5 Dense rivers pruning

The last step of the process is to detect the regions where rivers are too dense (rivers are too close to each other) and delete the less important of them. The algorithm that we developed uses buffers to find which rivers are too close to others and uses the same scoring procedure as before to assess the importance of a river before deleting it.

A buffer is built around every river course and the percentage P of the area of this buffer that is covered by other buffers is calculated and assigned to the river course (the percentage is the ratio covered area divided total area). All the river courses are then decreasingly ordered by the percentage P: the higher the value of P and the more this river course is close to other rivers courses. Starting from the highest P, the importance of every river course is evaluated and, if below certain parameters, it is deemed not important and it is deleted. When a river course is deleted, all the river sections comprising it are deleted, its buffer is deleted, and the values of P of the neighboring river courses updated accordingly. The process continues until the highest value of P is below a certain threshold $Pmax$ or there are no more river courses to be deleted over that threshold (i.e. all the river courses having P bigger than the threshold are deemed too important to be deleted).

After some tests, we found that good values for the parameters are: $Pmax$ = 50%, buffer size = 120m. A threshold on the values S, L and B of a river that is candidate to be deleted is used to decide whether the river is important or it can be pruned; the thresholds, found empirically, are the following: $S <= 2$, $L <= 1000$ meters, $B < 4$.

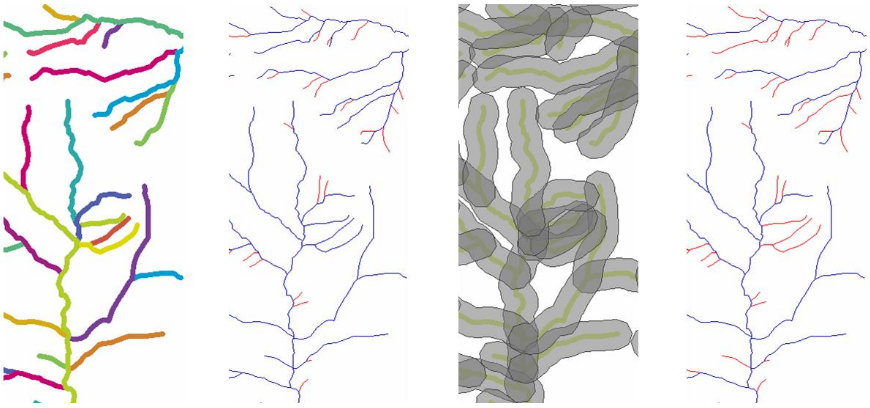


Figure 31: four steps in the pruning process; from left to right: river courses reconstruction, pruning by length, buffers to calculate the overlap, pruning by density

6.4 Generalization of ditches[9]

As the Oxford dictionary defines it, a ditch is “a narrow channel dug at the side of a road or field, to hold or carry away water”. Ditches are man made features used to convey water; they are typical of rural environments, where they may run along the roads or inside the fields, eventually dividing different crops. Ditches can be dug singularly, but most often are made in groups, for example as part of an irrigation or drainage system.

Being a man made feature, ditches usually show a regular pattern: groups of ditches often run parallel to each other, in straight lines with similar lengths and equally spaced.

From the cartographic point of view, ditches are represented as single lines that might or might not be connected to other features of the hydrology network; ditches usually do not take part in the hydrology graph and as they are usually not described with a rich semantic they are treated as a different feature class than rivers and canals.

In rural environments, characterized by the absence of dense road networks or settlements, the straight patterns of ditches are a prominent feature and as such they should be retained during generalization. It is interesting to underline though, that what is important to the description of the rural landscape is not the single straight ditch but the pattern of the group as a whole. Because of this, ditches lend themselves to be typified.

[9] This work was done with Matteo Zanon, University of Padua (see Savino et al., 2011a).

Figure 32: on the left ditches connected to other hydrography features, on the right ditches not connected; the latter type is the most frequent.

6.4.1 Selection vs. typification

One may wonder whether there really is the need of a typification algorithm to generalize ditches, or a simple selection algorithm could do the deed. From our tests we found out that the generalization of ditches can be accomplished with a simple selection algorithm, but that the results are not always good.

Figure 33: from left to right: original data, only one ditch present, only another ditch present, a group of ditches present. No matter which ditch is selected and which are deleted, it is only the presence of a group of ditches that allows to convey the concept of the pattern.

A simple example is the case of a group with an even number of ditches: selecting one ditch every two, the resulting generalized ditches will be equally spaced from each other (if this was true in the original data) but will not be distributed evenly on the space that was covered by the original ditches.

In general, a selection algorithm offer less possibility to represent the generalized data as it is constrained to keep the position of the objects that are not deleted. On the other hand, a typification algorithm that creates a completely new representation of the generalized objects could disconnect them. As we said, though, the odds of a group of ditches being connected to the hydrography network are much less than a group of an even number of ditches (50% of probability in normal distribution) and we evaluated more important a good pattern representation that its connectivity.

For this reason, and in general the ability of a typification operator to perform a better generalization of patterns, our choice was to try to develop a typification algorithm for the generalization of ditches. Moreover this gave us the chance to approach a more challenging research topic.

6.4.2 Related work

Typification and pattern recognition are closely related, as the first step of typification is to understand the pattern that should be kept.

Pattern recognition techniques have been developed for roads and buildings. [Heinzle & al., 2005] find grid-shaped structures in roads analyzing the nodes of the road graph; [Christophe & al., 2002] find alignment in buildings while [Anders & al., 2000] develop a parameter free cluster recognition algorithm that can be used as a preprocessing step of typification. Many papers investigating pattern recognition techniques focus also on typification. [Regnauld, 1996, 2001], uses a minimum spanning tree to cluster and typify buildings, [Sester & al., 2000] develop a typification algorithm based on Kohen maps, [Burghardt & al., 2007] apply mesh simplification to the solve the same topic. Most of the work on typification focuses on buildings, but there are also some examples of road typification in [Thom, 2005], [Xuechen 2010].

6.4.3 The algorithm

As we wrote in section two, not all the ditches of a dataset are part of a group: some ditches run isolated, following the course of a road or surrounding a field. The first step of our algorithm was then to find which ditches are parts of a pattern and which are not: this is done analyzing the direction of each ditch and then clustering them in groups.

Depending on the way the data was digitized, it can be hard to recognize a pattern; because of this the ditches have been preprocessed to ease the pattern recognition. During preprocessing, every ditch is divided in segments with the same direction Algorithm performing this segmentation task already exist [Plazanet, 1995][Dutton, 1999][Garcia & al., 2000] but the almost straight shape of ditches allowed us to set up a quite simple algorithm that measures the angle between three consecutive vertices and decides whether it is small enough to consider the three vertices almost in-line, or otherwise to split the ditch in the middle vertex.

At the end of the preprocessing step, all ditches have been divided in almost linear segments; for each of these segments the centroid and the average direction is then computed. For two segments to be in the same pattern they must have a similar average direction and their centroid should not be too far away. The direction similarity and centroid distance threshold are controlled by two parameters of the algorithm. In our tests for the generalization of 1:5000 scale data to 1:25000 scale data we found $\pi/24$ and 50m to be good values for them.

All the segments that are found to be part of the same pattern are then grouped together in what we call a “ditch cluster”. At the end of the process if a ditch cluster contains only one segment, this segment will be flagged as not to be typified, otherwise all the segments in the same ditch cluster will be typified together.

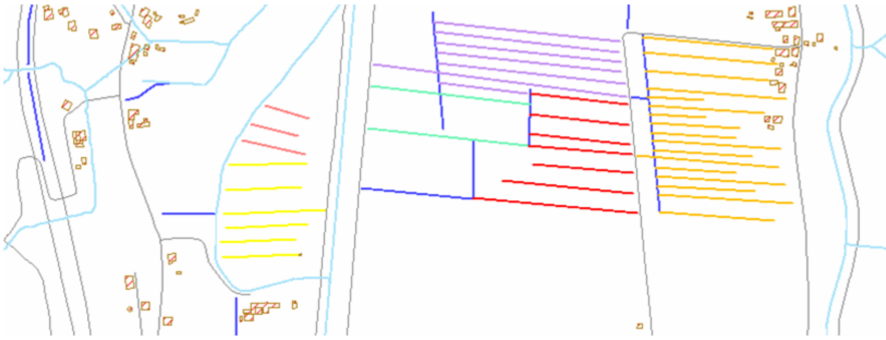


Figure 34: ditches grouped in clusters and ditches not to be typified (dark blue); each cluster has a different color.

Once we have identified the ditches to generalize and grouped them in ditch clusters, we need to typify them. That is to replace their representation with a new one, that is simpler (i.e. uses less geometries) but still conveys the same information of the former one.

The idea behind our algorithm is that the area interested by the presence of ditches should be considered as a “canvas” where we can freely draw the new ditches (as we will see later, this is not completely true and this behavior is constrained). The new ditches will run in the same direction of those that are replaced and will be equally spaced, accordingly to a distance parameter *SP* that is function of the target scale.

It is important to underline that the new ditches that will be drawn are completely new features that are not present in the original dataset: this is what makes this algorithm different from a selection algorithm.

The shape of the “canvas” where to draw the new ditches is obtained calculating a hull or envelope around all the segments of each ditch cluster. To obtain the shape of the “canvas” where to draw the new ditches we chose not to use the convex hull: since the convex hull may not follow closely the shape of the ditches, the resulting canvas could be too broad, enclosing areas that in the original data were not interested by the presence of ditches. Our choice was to be more conservative, in order to minimize the possibility of topological errors while drawing the new ditches: the algorithm that we developed will create a hull connecting all the end points of the ditches in each ditch cluster, thus obtaining a shape that both encloses all the ditches in the ditch cluster and minimizes its area extension. We call this shape “cluster envelope”. Each cluster envelope has its own centroid, boundary and direction - the average direction of the ditch segments it contains.

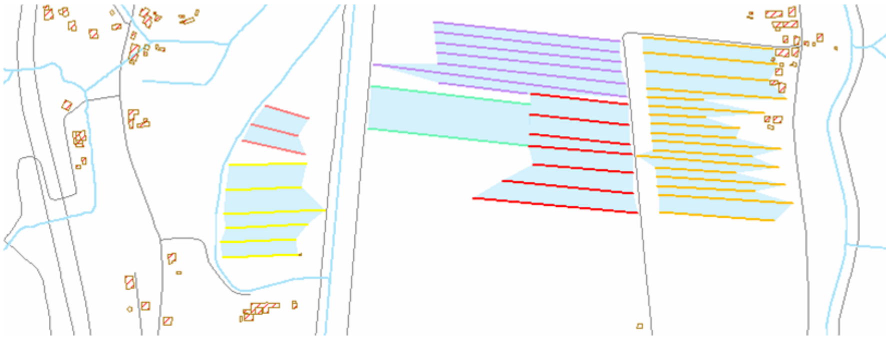


Figure 35: clusters of ditches overlaid with their cluster envelope (light blue)

Once the “canvas” has been set, the new ditches are drawn as following: a line *L* orthogonal to the envelope direction is traced by its centroid and on that a set of new points P_i, all equally *SP* spaced, will be created. All the new ditches will be drawn as straight lines, having direction equal to the envelope direction; for every line, only the part inside the envelope will be kept. Every line is drawn by a different point P_i on *L*. All the original ditches inside the canvas are then replaced by the new pattern of straight lines that are their new typified representation.

Once we know how to draw the new pattern of ditches, the only question left is to find where to draw it. As a first guess, the new pattern of straight lines will be centered on the centroid of the cluster envelope. To distribute evenly the new ditches over the surface of the ditch envelope, the number of point Pi contained in L is counted: if L can contain an odd number of points P_i, they will then pass on the centroid, otherwise they will pass at a distance *SP*/2 from it.

To avoid topological errors, this procedure should be extended to take into account eventual objects that are present inside the ditch envelope (e.g. a farm). In this case the center of the new pattern can be shifted along the line L to minimize the occlusion between these objects and the new ditches. Also object surrounding the new pattern can be taken in consideration: the position of the new pattern could be constrained by a buffer of size s on an object surrounding the ditch pattern (e.g. a road); this is equivalent as constraining the new ditches to a minimum distance from these objects.

Once that the presence of constraining objects is found, the algorithm tries to find a better position for the ditches: this is done with a simple iterative procedure that shifts *n* times the pattern along the *L* line of a measure *SP*/*n*, trying to minimize the intersection between the ditches and the constraining objects. The parameter *n* is set by the user and could be set as *SP*/*n* < *map accuracy*.

If a minimum different from 0 is found, the algorithm then uses these objects to crop the new pattern. To obtain a better result the crop can be done applying a small buffer around the objects, in a similar way to what is done in maps with the application of a halo around names or objects to emphasize.

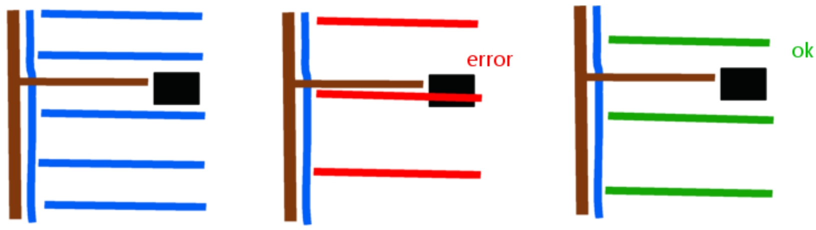


Figure 36: the placement of the new pattern is decided evaluating the position of eventual occluding objects, e.g. buildings

6.5 Generalization of big irregular areal features

In the GeoDBR model most of the feature classes have areal geometry; some of them represent small objects (as buildings) with regular shapes (e.g. straight boundaries, right angles), while other represent objects that have a larger extension and an irregular shape.

This section will describe the algorithms developed to handle big irregular areal features; under this name we group feature classes as woods, crop fields, lakes, quarries, glaciers to name a few.

The IGMI specifications allowed us to develop generalization algorithms that can be applied to many feature classes having this type of areal geometry: the algorithms described next might be considered as tools that are used in different cases. Not all the algorithms are applied to the same feature class: simplification is always applied, while aggregation and extension to linear boundaries mutually exclude collapse to line (features that are collapsed to line are aggregated as lines).

simplification
aggregation and selection
extension to linear boundaries
collapse to line

6.5.1 Simplification

The simplification of the areal geometries is achieved applying the Douglas-Peucker algorithm to the outline of the polygons.

Simplification allows to delete from the geometry those points that are useless to the representation of the objects at the target scale 1:25000.

Although Douglas-Peucker is not topologically safe, the big size of the features treated and the small threshold used guarantee that the output geometries are valid. Because the Douglas-Peucker algorithm may lead to a loss of connectivity among adjacent features (e.g. a lake and its inlet), this case is especially handled by dividing the outline of the polygon to be simplified into smaller lines, splitting it at the intersection with any adjacent feature; each line is then simplified using Douglas-Peucker and the resulting outline is obtained by dissolving again all the pieces together. Since the Douglas-Peucker algorithm does not move the first and the last point of each line it simplifies, it is guaranteed that the original position of the intersection points are left unaltered, thus retaining the connection with the adjacent features.

We found empirically that a very small threshold of 1 meter is a good tradeoff to reduce the number of points in each polygon avoiding at the same time topological errors and the creation of angles too sharp in the simplified geometry.

6.5.2 Aggregation and Selection

To aggregate nearby geometries we decided to use an approach similar to that described in [Mackaness & al., 2008]. In this work tree patches at large scale are generalized into wood patches at small scale; the methodology works by buffering (either enlarging or contracting) the source tree patches depending on their size, merging those overlapping and deleting those too small.

In our approach, we first enlarge all the patches using a buffer operation and then merging together those patches that now overlap; following we erode the patches by applying a negative buffer, in order to return the patches to their original size. Finally, if the IGMI specifications provide a minimum area size for the feature class, it is applied, deleting all those patches with an area size below the required threshold.

The process has the following beneficial effects on the data:

features at a distance smaller than the buffer size are merged together, thus reducing the number of features and increasing their average size; this applies also to holes, that may be filled by the buffer;
the outline of the features are smoothed (sharp angles are rounded by the buffer).

The process has also some drawbacks:

working with buffers increase the number of points in the polygon outlines (each vertex is replaced by an arc);
the shape of the resulting geometries may have lost adjacency with features bounding them (e.g. a fence).

Applying again Douglas-Peucker will reduce the number of points in the polygons (and if the threshold is smaller than the buffer radius, the smooth appearance will not be lost completely) thus overcoming the first drawback; the next algorithm will solve the second one, editing the outlines of the polygons.

Figure 37: using a buffer based expansion and erosion the woods are merged together; in the picture: original data (left), buffers (in pink, middle), and final result (right).

6.5.3 Extension to linear boundaries

Areal features like woods or crop fields are sometimes bounded by man made features like roads or fences or by natural features like rivers. The IGMI specifications require that in the case the distance between the boundary of the areal feature and one of such features is below a threshold, the boundary should be moved in order to be coincident to that of the feature.

The algorithm developed uses a constrained Delaunay triangulation (henceforth CDT) to create a new geometry between the existing boundary of the areal feature and the required new boundary: this geometry will be then added to the areal feature, extending it. The approach is similar to what is proposed by [Jones & al., 1995] for area aggregation.

The strategy is quite simple: a CDT is built using the points and the boundaries of both the areal feature class to extend (e.g. woods) and of the boundary feature classes (e.g. rivers, roads, fences). The CDT returns a set of triangles: for each of them we know both either if one of its three sides is on a boundary feature or on an areal feature and on which features each of its three vertices is. The algorithm searches the triangle set for all the triangles having one vertex and one side on an areal feature and on a boundary feature and adds them to the areal feature; the triangles are added only if the distance between these two features is below a threshold (the distance is computed for each triangle as the minimum among the height of the triangle and the lengths of its two sides connecting the two features). The search for the triangles to add is linear in the number of triangles in the CDT; to optimize the time necessary to create the new geometry, the triangles are not added one by one to the areal feature, but are merged into triangle strips and then added just once all together.

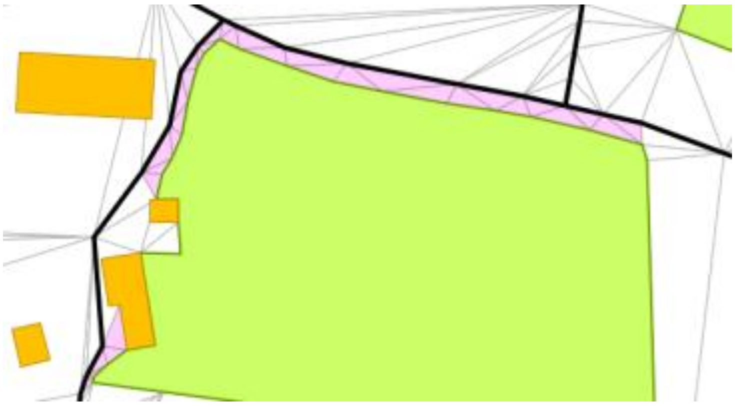


Figure 38: the extension of the boundary of a wood patch using the CDT (gray lines); woods are in green, roads in black: the pink triangles are those selected to be merged to the wood to extend its boundary; a threshold on the height of the triangles is used to select the triangles. Overlaps between buildings (orange) and woods are allowed by the specifications.

6.5.4 Collapse to line[10]

The operation to collapse an area to a line (also referred as the skeleton of the polygon) has been investigated and used by many (e.g. [Ai & al., 2002], [Cecconi, 2003], [Haunert & al., 2004]) and relies on the development of various algorithms (Medial Axis [Chin & al., 1995], Chordal Axis [Prasad, 1997], Straight Skeleton [Eppstein & al., 1999]).

We decided to use the approach proposed by [Bader, 1997], that uses a Constrained Delaunay Triangulation (CDT) to perform this operation. To build the skeleton, the algorithm first creates a triangulation using the points of the polygon P; the edges of the triangles created can lie on the boundary of P (constrained edges) or in its interior (unconstrained edges): on the base of the number of constrained edges that it has, every triangle is then classified in type 0,1,2,3. Following, for every triangle one or more skeleton edges are created: every edge connects the middle point of an unconstrained edge to another middle point, the centroid or the opposite vertex of the triangle.

[10] This work was done with Martijn Meijers, University of Delft.

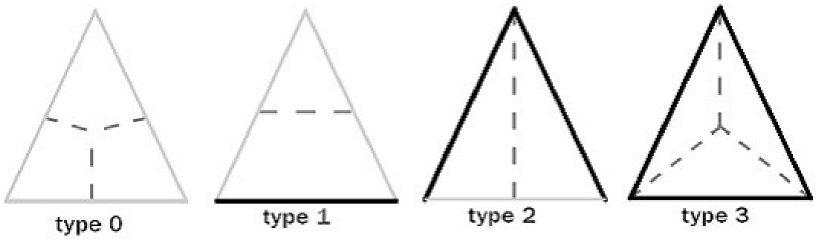


Figure 39: type 0,1,2,3 triangle. Constrained edges are in bold black, skeleton edges in dashed gray

The edges created (skeleton edges) are then dissolved together to form the skeleton of the polygon P: the skeleton is made of polylines composed of a sequence of points lying in the middle of the edges created during the triangulation.

The simple principle at the base of the algorithm is actually further complicated by the need to detect which triangles are inside the original polygon (only these triangles will be used to build the skeleton), to add some skeleton edges to connect the skeleton to the boundary and to deal with eventual holes.

The triangulation guarantees that the skeleton generated by the algorithm lies completely inside the polygon, but has the drawback to lead to a skeleton that is composed of many points and usually having a jagged appearance, eventually with sharp angles. To mitigate this, the algorithm described above has been modified in order to support a topological safe simplification of the skeleton.

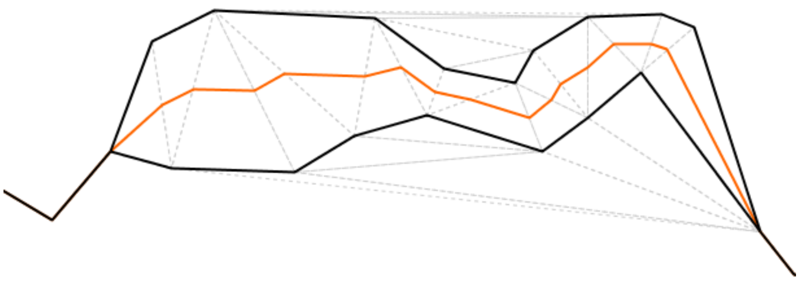


Figure 40: original polygon outline (black), and the resulting skeleton (orange).

The simplification algorithm is based on the Douglas-Peucker algorithm. The main idea behind our algorithm is that the points of every skeleton edge are not fixed, but may be moved by sliding them on the triangle edge they belong to: this degree of freedom can be used to simplify the skeleton. In fact, if a point p can slide on a position on the line between the previous point in the sequence p-1 and the following point p+1, the point p can be deleted from the polyline, thus reducing the number of vertices in the skeleton by one.

To ease the explanation, we will call these points lying on a triangle edge S-points (Skeleton points); every S-point lies on a triangle edge that we call T-edge.

In our implementation every S-point is a special object retaining the coordinates of the point and a pointer to each of the two endpoints of the T-edge it belongs to.

To implement our algorithm we had to modify our area to line collapse algorithm in two points:

first, during the creation of the skeleton edges, the S-points are created

second, just after merging all the skeleton edges in polylines, these polylines are simplified using the simplification algorithm

The simplification algorithm performs the following steps:

For a polyline composed of *n* S-points, being first point A and last point B:

1. draw a line from A to B: this is the baseline
2. cycle through the *n*-2 points between A and B and find the one furthest from the baseline, the S-point C
3. calculate I, the point of intersection between the baseline and the T-edge of C
4. if I doesn't exist or *C.dist(I) > tol* split AB in two polylines AC and CB and recurse on each of them
5. otherwise C is close enough to the baseline and can be deleted; continue checking the other S-points between A and B

Our implementation differs from the standard Douglas-Peucker implementation only in the step 3, 4 and 5. In these steps the standard DP checks whether simplifying the line will move the point C further than the allowed tolerance from its original position; our algorithm instead needs to check both whether the movement of the point C is within the tolerance and if this movement leaves it inside its bounds. The latter test requires computing the intersection between C's T-edge and the baseline (step 3). Our algorithm carries an overhead because it has to calculate an intersection instead of simply evaluating the distance. This adds to the worst case running time of the Douglas-Peucker algorithm constant value K, due to the calculation of the intersections, leading to a running time of k*n^2.

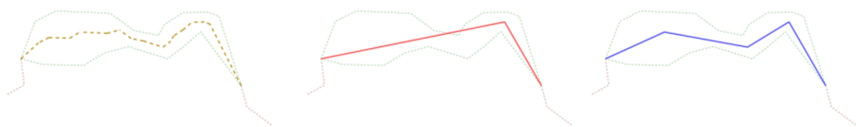


Figure 41: simplification of the skeleton. Left: original skeleton (dashed brown), middle: skeleton simplified with Douglas-Peucker (red), right: skeleton simplified with our algorithm (blue). Note that Douglas-Peucker algorithm lets the skeleton to cross the boundary of the polygon (dotted green)]

6.6 Generalization of linear features

6.6.1 Simplification and collapse of parallel lines

In the target data model utilities like power lines or pipelines are represented with linear geometries. The specifications require that in the case the lines belonging to the same feature class run in the same direction and at close distance, they should be collapsed to a single line. To abide to this requirement we designed an algorithm to collapse parallel lines (the adjective parallel is used here with a broad meaning: lines are not required to be perfectly parallel to be processed).

The idea at the base of the algorithm is quite simple: a buffer of radius R is drawn around each line and the eventual intersections between each buffer and the lines are found; R is set by the IGMI specification to 10 meters. Each intersection marks a section of a line that is closer than R to another line; since the proximity relation is symmetric, the sections are found in couples; each couple is then replaced by a single line. To create the single line the area to line collapse algorithm is used (a polygon is created from each couple by connecting the two lines together); the single line is then connected to the remaining section of the original lines with straight lines. It is still under evaluation how the algorithm should behave in the case of groups of linear feature being closer than R (e.g. power lines coming out from a transformer yard): the algorithm could replace the lines two by two or replace the whole group with only one line.

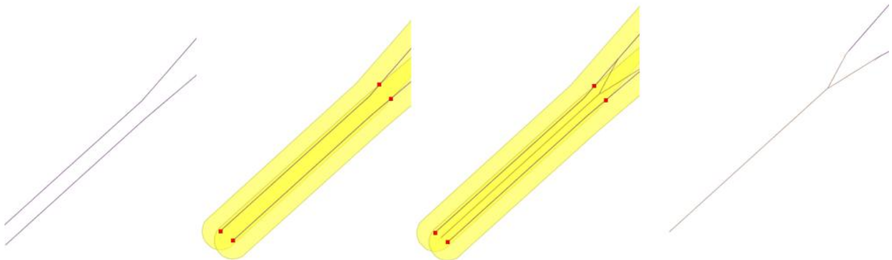


Figure 42: collapse of two parallel lines; from left to right: original lines, buffers (yellow) and points of intersection (red), the middle line is built and connected, final result.

6.6.2 Handling contour lines

According to the IGMI specifications, the generalization of contour lines is a straight forward process that requires only their selection based on their altitude (this is performed in the population step by querying via SQL the attribute storing this value). The source data in 1:5000 scale though has been digitized using a very high number of points leading to excessively accurate data for our target scale. The data is also excessively large in terms of memory space: in our sample dataset of mountainous terrain the contour lines account for about 370 Mb of data, while all the road network only for 7 Mb (dimension of the Oracle dump files); working with the contour lines is then very slow, also because their long, curvy shapes usually make vain the optimization of the spatial query engines.

It was our choice then to perform line simplification on the contour lines, applying the Douglas-Peuecker algorithm. Since contour lines do not represent any real world object, but just a symbol that is superimposed on the data to help to visualize the relief, they are not related to most of the other feature classes. The only noticeable exception to this is the hydrography network: as water tends naturally to flow to the points of minimum height, the path of the rivers follows that of the relief. As a consequence, it would be wrong to generalize either the contour lines or the river paths in such a way that the river paths do not flow anymore downward (e.g. in a valley) or crossing twice the same contour line.

In his work Gaffuri [Gaffuri, 2007] refers to this as the outflow preservation problem and describes a solution that working on the DTM of the relief in a multi-agent system adapts either the relief to the river, or the river to the relief.

In our case we decided to opt for a simpler strategy: by restraining the simplification threshold to be under the target data accuracy (2.5 meters) it was possible to reduce the number of points in the data virtually without affecting its accuracy; furthermore, keeping this value small, we could avoid that the simplification could make two different contour line to cross each other [Gokgoz, 2005]: since the DB25 has one contour line every 25 meters this would require a 1000% grade slope[11].

The position of the crossings between rivers and contour lines were kept fixed by dividing the latter on the intersection points with the former before applying the simplification; as Douglas-Peucker does not move the endpoints of the simplified lines, these points were not displaced. Furthermore, dividing the contour lines in smaller section helped to improve the running time for spatially querying this feature class.

Figure 43: simplification of contour lines; from left to right: original data, contour lines selected by height, simplified contour lines; it is possible to see that the rivers are still running in the thalweg after the simplification (right)

6.6.3 Fences and walls

In our source data, the GeoDBR, feature classes representing discontinuities (river banks, escarpments) or marking a division in the terrain (fences, dividing walls) are represented respectively by areal or linear geometries; the corresponding feature classes in the DB25 are all represented using lines. According to IGMI, during generalization all these feature classes are subjected to a reconstruction process aimed at closing eventual gaps in their geometrical representation. In other words, the IGMI specifications rule that a small discontinuity in the representation of one of the linear features above should be ignored and that the feature should be generalized as a continuous line.

For this purpose we designed an algorithm that "bridges" the gaps between linear features by adding a line between two of them in order to create a connection. As this algorithm is still under development, it will not be described in detail.

[11] The grade of a slope, as a percentage, is calculated as the tangent of the angle of inclination times 100 (Wikipedia http://en.wikipedia.org/wiki/Grade_%28slope%29, accessed 15/01/2011).

The algorithm differs between natural features and man made ones: as the former usually have an irregular shape, they will be connected using simple line segments, whilst to connect the latter also line with right angles will be allowed.

In both cases the algorithm will first create a graph structure made of nodes and edges from the original linear data: to each point of the original lines corresponds a node in the graph. Each node that is reached only by one edge is a candidate to be connected to another node by the creation of a new edge that will span the gap between the two linear features to which the nodes belong. For each candidate, a spatial query will return the nodes within a distance *D* equal to what the IGMI specifications state as the maximum gap that should be ignored: these nodes are too candidates to be connected.

The connecting edges between the first candidate node and those found by the spatial query are compared: those connecting to the original linear features at an angle too obtuse or too acute are discarded and among those remaining, only the shortest is kept.

In the case of man made features, before comparing the connecting edges, the algorithm evaluates if it is possible to connect the candidates extending the linear features to which they belong to: only if this is not possible the algorithm tries to connect the two candidate nodes as explained above.

The extensions to the linear features are drawn as line segments of length *D*, that are applied to the features on the candidate nodes and are collinear to the features in the point of application; an extensions is drawn only if the candidate node is an end point of the linear feature.

Figure 44: extension of walls using a triangulation (CDT): buildings in yellow, walls in black, roads in brown; the dashed gray lines are triangle edges created by the CDT. Green lines are triangle edges that can be used to fill the gaps; in the case the triangle edge is not collinear with the two walls it connects (red lines), the algorithm tries to connect the two walls by an L-shaped segment (purple lines); edges crossing buildings or roads are considered invalid.

If the extension intersects either another linear feature (or itself) or the other extension, the two line segments between the two candidate points and the intersection point are used to “bridge” the gap and are added to the graph as new connecting edges.

The connecting edges created are tested to not cross any relevant feature class: edges connecting river banks should not cross any river, or those connecting two walls should not cross a road.

In the case of walls and fences the generalization algorithm performs also a selection, in order to reduce the number of features stored in the DB25 to only those more relevant.

The algorithm uses the graph as a planar partition and computes the faces: each face represents an area enclosed by a fence. To assess the importance of these areas their area size is measured and they are classified on the base of what the area contains: small areas containing residential buildings are flagged as not important and their fences will be likely discarded; on the contrary, fences enclosing relevant features will be retained (e.g. fences enclosing a hospital, an airport, a big factory, one or more crop fields, a quarry, a mine, a dump, a power station, a transformer yard, a water treatment plant, a campground, a sport complex, a big isolated house, ...). In the case adjacent faces enclose the same type of feature, they are merged together and only the fences on the boundary of the resulting polygon will be kept.

6.7 Generalization of railroads

Among the networks on a map, railroads represent the less complex one: they are by far less numerous than rivers and roads, and their path is quite regular; by construction the railroad network is comprised by long and straight sections with few intersections and no abrupt changes of direction. The generalization of the railroad network is then usually solved by collapsing to one single line eventual sections of the network where pair of railroads runs parallel; to accomplish this it is possible to apply the algorithm for the collapse of parallel lines (see 6.6.1). At large scale, though, railroad data can comprise much more detail leading to very complex representations, especially around train stations (see Figure 45) [Cecconi, 2003].

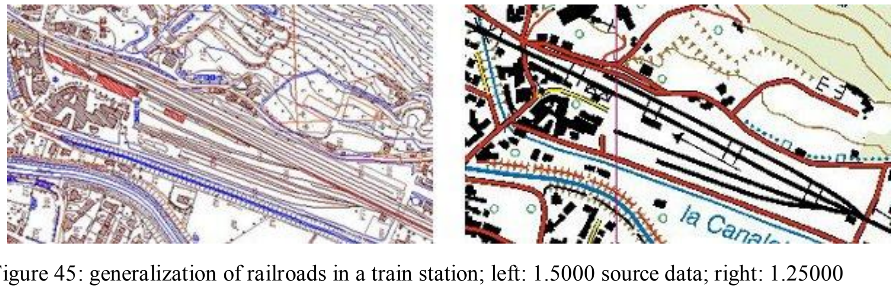


Figure 45: generalization of railroads in a train station; left: 1.5000 source data; right: 1.25000 hand generalized data

The generalization of rail stations is a typical problem of typification: it is necessary to draw a new representation of the railroad tracks that is simpler (i.e. it comprises fewer objects) but conveys the same information (i.e. the number and direction of the generalized tracks should still give the idea of a station).

This problem has not been completely solved yet.

The first algorithm that we devised to generalize the railroads relied on the application of the "good continuity" principle to identify the main tracks running through a station and then select the other tracks based on their distance from the main ones. This approach led to correct results in some cases, but could not handle train stations where there was not a main track. A second approach tried to approach the problem collapsing iteratively pairs of tracks to single lines. The algorithm first identified those edges of the railroad graph closer than a threshold (set to 20m) and then classified each couple as "circle", "triangle" and "square", applying to them different operations:

in a circle couple the two edges are connected at both their endpoints; they can be replaced by their middle line with no further operations;

in a triangle couple the two edges are connected only on one endpoint: they are replaced with a middle line that runs from this endpoint to the point in between the other two endpoints: the middle line should be then connected to all the edges that were connected to these two endpoints;

in a square couple the two edges do not touch each other: they are replaced with a middle line that needs to be connected to all the edges that were connected to the two that have been collapsed.

This second approach was able to generalize all the train stations in the test dataset but further tests revealed that it was not capable of handling much more complex inputs (see Figure 46).

At the moment we are looking for new approaches: we are currently investigating the use of a triangulation to draw the middle lines (see 6.5.4); to date the generalization of railroads needs further research.

Figure 46: example of different complexity in the source 1:5000 data; a simple case comprising only 27 elements (left) and a complex case comprising 288 elements (right).

6.8 Generalization of points

In the generalization process, the generalization of points is required only by two feature classes, precisely CA030 Spot Elevation and EC030 Trees; other feature classes have point geometry in the DB25 but they are mostly the result of a collapse to point of feature with polygon geometry at the source scale and thus do not need further generalization.

The generalization on point is basically a selection: both the GeoDBR feature classes corresponding to the DB25 feature classes CA030 and EC030 in fact comprise a big number of points that should be reduced to fit a point density proper of the target scale.

Before being processed by the algorithm, the points are first filtered: trees that are covered by elements of the feature class EC030 Trees (polygon) and spot elevations that are covered by those of the feature class CA010 Elevation Contour (line) are deleted.

The point selection algorithm has been implemented on the base of the work on mesh simplification done by Cecconi [Cecconi, 2007]. Starting from other works on mesh simplification [Turk, 1992, Hoppe et al., 1993], the author develops a simple strategy to reduce the number of vertices of a triangular mesh applying iteratively an edge collapsing operation [Hoppe, 1996]. In the work of Cecconi, the vertices of the mesh are the center of gravity of a group of buildings and the simplified mesh is used to obtain a typification of the group; at each iteration the algorithm collapses the shortest edge of the mesh, reducing by 1 the total number of vertices.

In our algorithm a similar approach is applied to the simplification of trees.

First a Delaunay Triangulation is built on the points of the feature class, then starting from one vertex of the triangulation the length of all the edges departing from it is calculated: if one edge is shorter than a threshold, it is collapsed, the triangulation is updated and the algorithm iterates on the new vertex; if none of the edges is below the threshold, the algorithm iterates on one of the neighboring vertices. When the process ends, all the points closer than the threshold have been deleted.

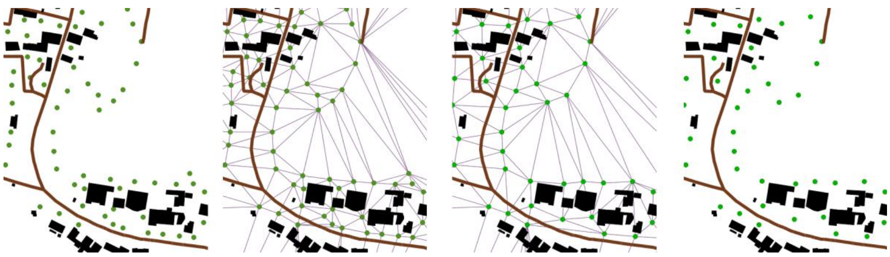


Figure 47: selection of trees (in green) using a mesh simplification technique; from left to right: original data, CDT built on the original data, CDT after the edge collapse, final result.

The selection of the spot elevation points is slightly different: since these points represent a precise spot on the surface of the terrain, it is not possible to apply the edge collapse, as it will move the spot away from its original location (e.g. moving the peak of a mountain). Thus the previous algorithm is slightly changed: once two vertices of the triangulation too close to each other are found, one of the two is deleted only if the difference in height between them is below a threshold; the vertex deleted is always the lowest one. The algorithm tries to delete points that are "similar" (i.e. close and not representing a relevant change in height) while preserving the most "noticeable" (i.e. the highest).

6.9 Operators

In the following table the generalization algorithms described above have been classified and grouped as operators according to their functions; the first column keeps the names of the operator (after [McMaster and Shea, 1989], the second column refers to the type of data to which each operator is applied to, the third column refers to how the operator was applied on the data or to what purpose and the forth column points out whether the algorithms implementing the operator are a novel contribution of this thesis or not, in the latter case citing the most relevant works used to derive them.

Operator	Used for	Function	Note
Simplification	Buildings		Derived from [Sester, 2000]
	Polygons and Lines		See [Douglas and Peucker, 1973]
Collapse	Polygon to line		Derived from [Bader, 1997]
		Topological simplification of skeleton	Original
Smoothing	Woods, crop fields		Derived from [Mackaness & al., 2008]
Enlargement	Woods, crop fields	Extension to neighboring linear boundaries	Derived from [Revell, 2005]
Exaggeration	Buildings		Proposed by [Sester, 2000]
	Hydrography	Harmonization area/line transition	Original
Squaring	Buildings	Angle squaring	
		Square-ability test	Original
Selection	Hydrography	By length	
		By density	Original
	Buildings	By size and isolation	Derived from [Boffet,

			2001]
	Road network	By length with test on access to building	
		In road junctions	Original
		Misclassified highway segments	Original
	Fences		Original
	Points		Derived from [Burghardt and Cecconi, 2007]
	Woods, crop fields		
Amalgamation	Buildings	Fusion	
		Merge	Original (the process is intended to be followed by simplification)
	Woods, crop fields		Derived from [Mackaness & al., 2008]
Typification	Ditches		Original
	Silos		Original
	Railroads	In stations/train yards	Original (under development)
	Toll-plazas	In highways	Original
Classification	Road network	Harmonization	
	Road junctions	Detection and classification	Original
	Highways	Classification of components	Original
	Hydrography	By width	Original
		Flow reconstruction	Original
		Data enrichment	Original

Table 2: the algorithms developed grouped as operators

6.10 Some final remarks

6.10.1 Selection and classification

The table shows that the two most used operators are selection and classification. These two are in fact the most important operations in a generalization process like ours: due to the small scale gap between the source and the target data the reduction of information could be performed on a single object base (i.e. simplification and selection), with more complex operations on groups of objects (typification) being restrained to just some cases (e.g. ditches). Furthermore, in a process aiming at the generalization between two geographical databases, operations dealing with problems related to the context of map derivation (e.g. displacement) could be not considered.

Indeed in our process selection and classification are closely related: while selection actually drives the generalization process, deciding whether and how an object in the source database should be translated (both semantically and geometrically) in the target database, classification enables the selection process to make correct choices (e.g. telling what is important from what is not).

6.10.2 Data enrichment

It is important to note that the classification operator relies not only on the source semantic data, but actively collects more information in a process of data enrichment (e.g. see [Neun et al., 2004]). As a precise design choice most of the classification algorithms that we developed rely on morphologic analysis to gather such further information; three factors mainly influenced this decision:

the first was a matter of necessity, as we could not access to any other data source that could provide the needed information;

the second was the desire/aim to develop algorithms that were robust to semantic error: as a matter of fact input datasets may contain both semantic and geometric errors; while geometric errors are more easy to detect and to some extent to correct (e.g. with a topological validation), semantic errors are more subtle. Developing an algorithm that relies more on geometric data than on semantic data allows to generalize even input datasets that contain some classification errors;

the last was the challenge to mimic the human capability of reading a map: if the human eye could easily detect -for instance- a complex road junction, could we develop an algorithm with similar ability?

In most of the algorithms developed, data enrichment was a key factor to obtain a good generalization.

6.10.3 Use at different scales

Although the algorithms that we developed have been conceived to solve our specific generalization problem, some of the generalization strategies devised can surely find application in more general contexts; in particular we believe that with appropriate parameters the algorithms for the generalization of the hydrography network and the typification of ditches and silos could be applied to different scale ranges (with the typification of silos being possibly extended also to buildings, as their representation becomes more regular due to symbolization); on the other hand, the generalization of the road network is probably applicable only to large scale source data, as at smaller scales the representation of the road network is usually already simplified (e.g. complex road junctions are not present).

As the CARGEN project has been extended to investigate the generalization to the 1:50000 scale, all the process is currently being tuned and tested to produce data at this smaller scale; early results are confirming our presumption on the adaptability of the algorithms developed.

6.11 Conclusions

In this chapter most of the algorithms developed for the CARGEN project have been explained and those just designed have been outlined. The generalization algorithms use both known and new techniques, providing some original contributions to the research on cartographic generalization. Modeling the algorithms as operators it was shown that most of them deal with the implementation of selection and classification: these proved to be the most important operators to perform a DB to DB generalization at our scales. Selection and classification act in tandem, with the former reducing the data and the latter driving the process; data enrichment is a key factor to perform classification: most of the classification algorithms rely on morphologic analysis to enrich the data and overcome eventual classification errors in the source. At the end of the chapter it was suggested how the algorithms developed, that have been conceived pursuing the generalization at 1:25000 scale, might as well be applied to generalize smaller scales.

Chapter 7

Results and evaluation

7.1 What is a good generalization?

The evaluation of the results is a very important part of each process and cartographic generalization is of course not an exception to this. But to wonder whether the process produced a good generalization is as spontaneous as it is difficult to define what a good generalization is. As generalization is some how closer to an art than to a technique, it is difficult to formalize what is a good generalization and what is not, and this makes difficult to evaluate if the results obtained meet the expectations.

To develop a process able to automatically assess the quality of a generalized result might be as complex as it is to develop the very same generalization process; actually in some cases the former is part of the latter (e.g. see [Agent, 2000]).

In this chapter the different type of errors that can occur during generalization are introduced, followed by the explanation of how our process deals with them. Next the results of the process are shown and commented: first those of each step and then the global ones.

7.2 Generalization errors

As the generalization process can be divided in the two processes of semantic generalization and geometric generalization, these two can be also identified as the possible sources of errors in the process. In assessing the quality of the data produced by a cartographic generalization process in fact we can identify two main type of errors:

semantic
geometric

The first type of error is that affecting the semantic data, and is related to a wrong "translation" of the source data to the target data model: a feature in the source data is generalized to a feature that has a different definition or different specifications. Because source and target data models do not usually match perfectly, the definitions in the two models will always be different; to incur in a semantic error though it is necessary that the definition and specifications of the source feature and the target one are not semantically equivalent or the former is not comprised in the latter.

This type of error may affect the choice of the target feature class, or an attribute value; in general these errors occur because of mistakes in the semantic generalization process. An example of this is a mosque being generalized to a church: this might prove wrong if the definition of church recognizes strictly the Christian churches, but it is correct if it has a broader meaning, encompassing a general place of worship. Semantic errors may also derive from geometric errors: for instance if the size of a river is assessed incorrectly, this river may be generalized in the wrong feature class.

The second type of error affects the geometries of the generalized data and they are accountable to errors in the implementation of the generalization algorithms. This type of error can be further classified as:

invalid geometries
geometries not complying to the specifications
geometries with topological inconsistencies

The first type of error is caused by geometries not complying to the OGC Simple Feature Specification, as a polygon with self-intersecting boundary.

The second type affects those geometries that according to the specifications should not exist in the target database; an example is a polygon representing a quarry whose area is smaller than the minimum area size required for this feature class or two objects closer than the minimum distance threshold prescribed.

The third type of error occurs when geometries violate topological constraints. The term topological constraint is used here to refer to any rule that enforces a spatial relationship between a feature and another, for instance that a building and a lake can not overlap. In general topology rules allow to map real world concepts into the data (e.g. that a road can not be over a river if it is not also over a bridge). Through these rules it is possible to formally define the relationship among all the features in a database, as the constraints might be extended also to describe relations among disjoint objects (e.g. a building should not be closer than 5 meters to a railroad).

Topological errors may occur if the source and target database implement opposite topology rules (e.g. one allowing road to pass through woods while the other requiring the woods to be cut in correspondence with a road) but this is more of a model related issue; the most relevant source of topological errors are the generalization algorithms that, modifying the geometries, affect the relations that they have with the neighboring objects.

There is another type of error that jeopardizes the process of cartographic generalization. The output of the process might fit perfectly the definitions, comply with all the specifications and not contain any invalid or topologically wrong geometry, but the data may still not be a good generalization due to some defects.

We will call these errors cartographic errors: a cartographic error occurs when the data at target scale does not represent a good generalization of the source data.

It is not easy to define what causes a cartographic error, as it is not easy to define what a good generalization is or, in general, what is a good map. The aspects related to cartographic quality, as legibility, information convey and in general the ability of the map to be a good source of information for the user, are not easy to formalize: the same reasons that make an automated cartographic generalization process hard to be developed make difficult to evaluate its results. Because the cartographic quality is closely related to representation, cartographic errors are most likely caused by wrong or incomplete geometric generalization algorithms. Some of the cartographic errors might be modeled as topological errors (e.g. the generalization of a road makes it to overlap a nearby lake), but usually they affect characteristics of the information that are not modeled in any of the other type of errors (e.g. the generalization of a group of buildings makes them to lose the spatial pattern characterizing their original distribution).

7.3 Handling errors in the CARGEN project

Errors in the CARGEN project are dealt with mostly in an implicit manner.

The occurrences of semantic errors in the generalized data have been reduced to zero during the process of model generalization: the manual analysis of the two source and target data models and their alignment led to a precise mapping that has been successfully tested to provide a correct semantic generalization.

Geometric errors are handled within each algorithm and each step of the process. Invalid geometries are detected and corrected using functions commonly available in spatial DBMS or in the JTS, while the specifications on the geometries are enforced by the generalization algorithms as their first task.

About topological errors, the approach in the development of the algorithms has always been keen on minimizing the chances to cause this kind of errors: for example the simplification algorithms are applied with small thresholds and eventual new geometries are placed carefully in areas that are free of any obstructing object. As the DB25 implements only a short list of topological rules, all of them being completely compatible with those of the GeoDBR, our approach relies on the conservation of the topological relations of the source data.

In the generalization process each step is responsible of the quality and correctness of the data it produces, but a step for the automated global evaluation of the results is missing. Although models and frameworks for the evaluation of generalized data exist [Mackanness and Ruas, 2007], they would require a complex implementation and further research. Our choice was then to focus on the development of the generalization algorithm and to rely on visual inspection to evaluate the generalized data: this required only a GIS (e.g. OpenJump) and allowed to speed up the development of the algorithms, especially at the beginning of the project; furthermore the sample dataset and those used to test the algorithms were reasonably sized to perform the evaluation quickly.

To ease the process of visual evaluation of the data, we developed a plug-in for OpenJump that symbolizes the layers using the official DB25 legend issued by

IGMI. Although the output of our process is a DLM and not a DCM, applying this symbolization allows to distinct at a glance the various feature classes and detect areas of the map where the generalization quality might be insufficient.

7.4 Test results

The sample dataset, comprising a 132000 hectares territory belonging to the "Parco delle Dolomiti Bellunesi", has been used as a test area for the development of the whole process; the dataset comprises both mountainous and plain territory, with medium and small sized settlements and its extension corresponds to about 10 IGMI map sheets at 1:25000 scale. In some cases the algorithms have been tested also on other cartographic data (plain and hilly areas and costal areas), to gather further evaluation of their behavior.

In this section the results on each step of the generalization process are presented and commented.

7.4.1 Time performance

The process designed is not bound by any real-time constraint thus since the beginning time performance was not considered a priority; nevertheless, the running times of the algorithms have been monitored during the development as it is our belief that an algorithm that takes too long might not be using the correct optimization, or be too complex (and complexity is a hideout for software bugs).

To generalize the sample dataset took to the process 20 hours. The time was recorded on a middle-range machine, an Intel Dual-Xeon 2.66 GHz with 3Gb of DDR2 RAM, SCSI 10000 rpm hard drives in RAID 5 configuration, running Windows Server 2003, Oracle Spatial 10.2g and Java 1.6.

The most demanding steps of the whole process were (time in hours:minutes):

Building generalization 4:50 (around 68K buildings)
Road network (ordinary) generalization 4:25 (around 22K edges)
Hydrography generalization: 2:57 (around 10K edges)

Most of the algorithms developed rely on the API of Oracle Spatial to handle both the spatial query functions and the geometry editing functions. This behavior shifts the workload onto the Oracle server that, despite being very performing, especially on large spatial queries, suffers from the continuous transits on the tcp/ip socket, especially for small queries on single objects.

The running times could be diminished adopting a more widespread use of Java code (e.g. Java Topology Suite) to handle the operations on single objects or small groups of them: this would avoid the continuous queries to the DBMS, eliminating both the overhead due to the tcp/ip socket and to the SQL. Test run using new algorithms, designed following this idea and using new libraries, show that they benefit of faster running times.

7.4.2 Step by step results and future developments

7.4.2.1 Population of the target database

The process, customized on the two data models, produces correct results.

It is important to remember that this has been made possible by aligning the two data models, changing each of them in order to increase their compatibility.

7.4.2.2 Generalization of hydrography

The results have been visually inspected by expert cartographers and found generally good.

The flow direction reconstruction algorithm worked very well but in one of our tests it could not provide a complete reconstruction due to the high number of flat and uphill sections. Despite some little improvements that can be made on the algorithm, this proves once again that to have a good generalization it is important to have a good input data: in the absence of a directed graph (i.e. having the flow direction of the streams) the z coordinate of the data must not contain errors. The reconstructed river courses show in some cases sharp bends: it is under investigation the use of "good continuity" as a parameter in the river course reconstruction algorithm (e.g. see [Touya, 2007]).

The pruning algorithm was found to perform well, even though the speed of the density pruning could be probably improved. The pruning solved also some braided streams sections removing the shorter branches.

The generalization of braided sections of rivers need further investigation, as the first attempts did not produce any satisfactory result; the present direction of development is to detect the area enclosed between the branches of a braided section of the river and typify the clusters of these areas either by collapsing each area to a line, or by merging neighboring ones.

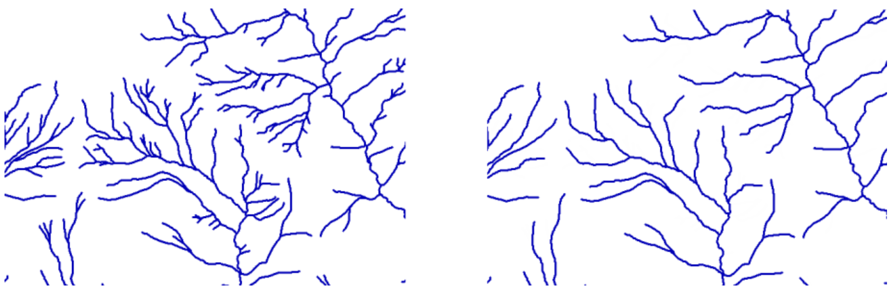


Figure 48: results of river generalization: before (left) and after (right)

7.4.2.3 Generalization of the road network

The algorithms proposed to enrich the road network, detecting and classifying road junctions and the different parts of the highway showed to produce consistent results on different datasets.

In general the generalization of the road network provided good results, pruning the network without generating errors.

The process is, in some cases, too conservative, retaining too many little roads, especially inside residential areas. Another aspect that could be further developed is the extension of the type of road junctions recognized by the algorithm; in particular roundabout connected to other roundabouts.

An aspect that had not been taken in consideration is the generalization of parallel ordinary roads that at the moment are not treated: they could be collapsed to a single line applying the same procedure used for the dual carriageways of the highways.

The harmonization of roads produces interesting results: the approach could be extended also to small roads, relaxing the “good continuation” condition, to improve further more the results; also, it would be interesting to enrich the source data in order to have a finer classification to use to operate a selection of the roads, as it is done with the hydrography. At present we are investigating the use of the metrics proposed by [Jiang & al., 2002].

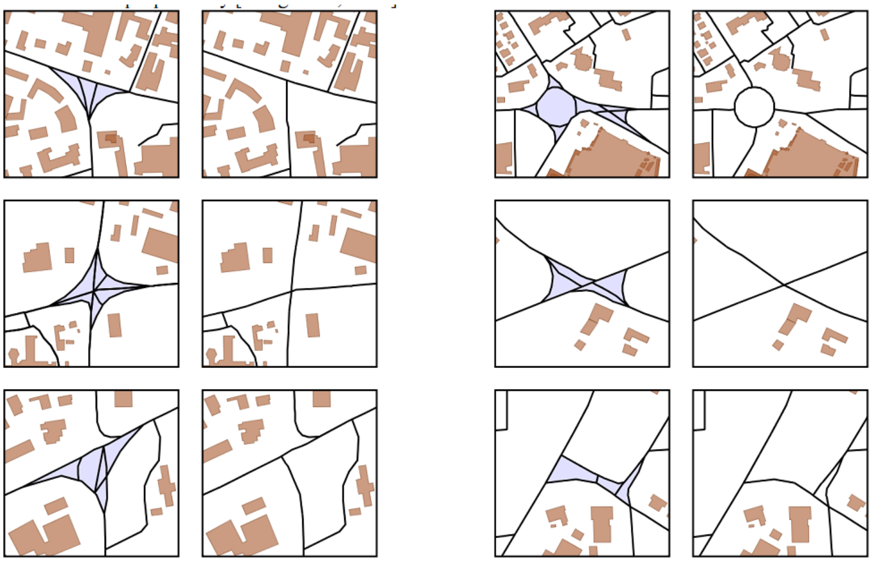


Figure 49: some results of the generalization of road junctions. From left to right, top to bottom: T-junction, roundabout, crossroad, paired T-junction and unclassified junctions. In the case of unclassified junctions the algorithm provides a “best effort” solution.

Figure 50: results of the generalization of the highway: original data (top), generalized data (bottom); it is possible to observe how the dual carriageways have been collapsed to a single line, how the slip roads have been connected to this new line and how the toll-plaza (bottom left corner) had been simplified.

7.4.2.4 Generalization of railroads

Still under development.

7.4.2.5 Generalization of buildings

The process was tested on more than 80000 buildings, producing convincing results.

One of the aspect to improve is the behavior of the algorithm with two adjacent buildings that can not be merged together (e.g. because belonging to different feature classes): in this case the geometric operations on each single building can lead to topological errors.

As the case of adjacent building that can not be merged is not rare, a procedure to correct these topological errors has been developed: the algorithm performs a difference between any pair of overlapping buildings, thus removing their intersection. This approach, albeit effective in resolving the issue, may lead to the formation of slivers and, in general, to an irregular shape of one of the two buildings. At present we are developing a new approach that handle adjacent buildings as a set of edges and nodes instead that a set of faces; in the future all the building generalization algorithms should be modified in order to be applicable to the single edges.

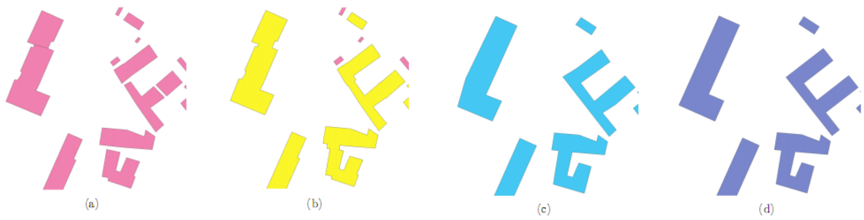


Figure 51: generalization of buildings. a) original situation, b) selection and aggregation, c) simplification, d) squaring

7.4.2.6 Generalization of ditches

The algorithm was tested on two datasets, one comprising a mountainous territory, the other a plain, the latter being more rich of ditches. In both cases the algorithm produced good results, reducing the number of ditches and creating uniform patterns. The algorithm is able to identify ditches that are part of a pattern and those that are not, and to create a new representation for the ditches that should be typified.

The algorithm produced also some errors:

- a couple of ditches running along the same road might be clustered and typified together, with the new pattern being comprised of only one new ditch, running exactly over the road among the original ditches; such a situation is not so uncommon and while the present algorithm will crop the new ditch, a better solution should be found.
- when the ditches in a group are connected to each other by another ditch or hydrographic element (river, canal) if the shape of the ditch envelope doesn't follow the shape of this connecting element, in the new pattern some ditches can be disconnected from it.

The algorithm also leaves space for future improvement and development:

- in some cases the typification results in a representation that is too uniform all over the dataset, as the algorithm applies only one fixed measure (50 meters) to typify all the ditches; a possible solution could be to use instead two thresholds, one to guarantee a minimum distance between the ditches, another (bigger) to prevent the typification of ditches that are not dense enough; inside this range the algorithm could apply a “proportional” typification, i.e. the ditches in a typified cluster are drawn at a distance that is a fraction of the average distance in the original cluster
- in some other cases instead a major uniformity is desirable: in the case of ditch clusters that are spatially close to each other (e.g. clusters separated by a road)

and have similar average direction it should be possible to apply the same typification to all of them (hence working on clusters of ditch clusters).

Figure 52: generalization of ditches: before (left) and after (right); the number of ditches has been greatly reduced, while the patterns have been maintained.

7.4.2.7 Generalization of large areas

Still under evaluation

7.4.2.8 Generalization of linear features

Still under evaluation

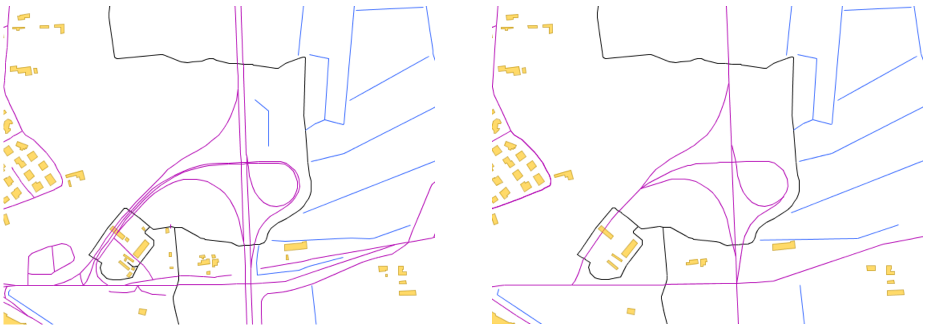
7.4.2.9 Generalization of points

The algorithm needs further tests to find the correct distance value for the selection of the points. In the case of the trees, applying only one threshold leads to a reduction of only the closest points: it would be probably better to use two thresholds (as proposed for the ditches) and to apply the selection also on the base of the local density. Furthermore in some cases trees should be typified and not only selected, as their original placement follows a spatial pattern: this is the case of trees running by a road or a river; in these cases the generalization should try to maintain the pattern: groups of collinear trees should be found and, if running along another linear feature (road, river, railroad, …) make then the edge collapse happen on this line.. For the selection of elevation spots we are implementing and testing an approach similar to what has been proposed by [Baella et al., 2007].

7.4.3 Global results

Although some steps of the process are still under development, the results obtained are promising. While a more formal evaluation is still being carried on by the IGMI and the Regione Veneto, the project will continue in its development, improving and extending its capabilities.

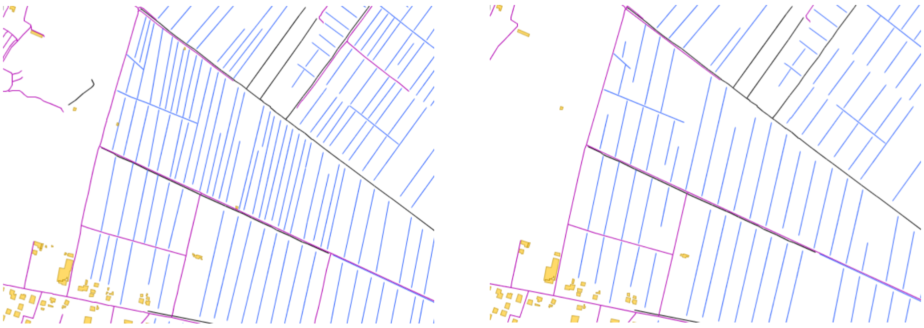
Because not all the algorithms designed have been completely tested and developed, the results that will be shown next will contain just some of the feature classes comprising the GeoDBR and DB25. If not differently stated, in the following examples the source data (1:5000 scale) is depicted on the left, while the generalized data (1:25000) on the right. The images display the feature classes corresponding to rivers (black lines), ditches (blue lines), roads (purple lines), contour lines (green lines), woods (gray polygons) and buildings (orange polygons).



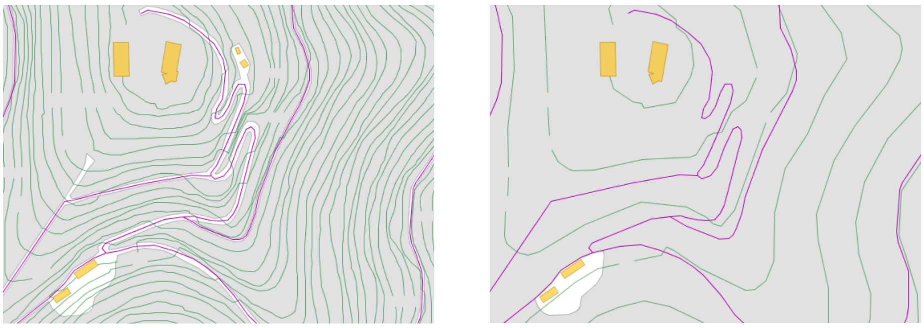
In the figures above it is possible to see that carriageways have been collapsed to a single line, the slip roads and the toll-plaza have been simplified, some minor roads have been deleted; also some smaller buildings have been deleted and some short branches of the hydrography network.

In the figures above it is possible to see how the road junctions have been generalized and how both some smaller roads and buildings have been deleted.

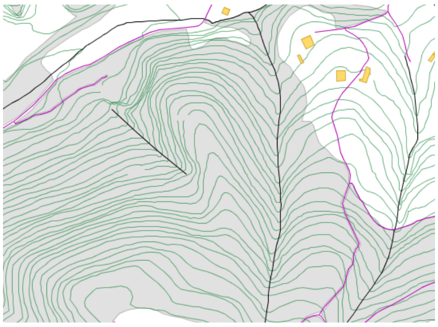
In the pictures above it is possible to see the effect of the generalization of the buildings: smaller buildings are deleted, the outline of buildings is simplified, small juts are removed and the resulting geometry is squared.



In the pictures above it is possible to notice how the number of ditches has been reduced; it is also possible to see how the density of the pattern resulting from the typification is constant for every group of ditches.



In the pictures above the wood has been generalized: small gaps in its coverage have been filled and its boundary has been extended to the nearby roads. It is also possible to notice the generalization (simplification and elimination) of buildings.



In the pictures above: despite the simplification of the contour lines the original accuracy by which the rivers flow in the thalweg is not compromised. The pictures show also that it is very unlikely for two contour lines to cross after simplification, even in the presence of a steep slope. In the pictures it is possible to see the reduction of contour lines, roads and short rivers.

Next we will show how an excerpt from a DB25 paper map and the generalized data compare; the data has been symbolized using the OpenJump plug-in that we developed. From top to bottom, next it will be shown: the source data, the generalized data, the symbolization of the generalized data and a part of a DB25 paper map.

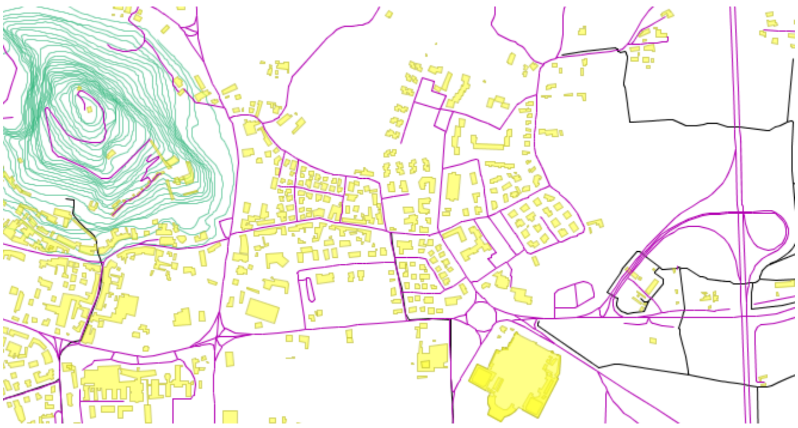


Figure 53: source data at 1:5000 scale.

Figure 54: from top to bottom: generalized data at 1:25000 scale, generalized data symbolized and an excerpt from the DB25 paper map.

Comparing a generalized map with an existing one is probably the easiest way to assess the quality of the generalization process. Although the result of our process is a geographical database and thus it is not intended to produce a map, comparing the generalized data with a paper map can be nevertheless very useful to evaluate the performance (in terms of cartographic quality) of the process.

From the last two images above we can see that the generalization of the roundabout and of the road junctions along the street heading east-west at the bottom of the picture is pretty good and resembles closely that of the map; also the generalized buildings and highway match very well the hand-drawn data. On the other hand, it is evident that the generalization of the road network still needs improvements, especially for what concerns the classification of the data: some roads have a different classification from that of the map, leading to a more clogged representation; classification differences are visible also among the buildings.

Analyzing an existing map not only is an easy way to evaluate the generalized data, but can also be a valuable source of information both to design a generalization algorithm and to tune its parameters. It is often the case to realize that the map embodies many rules that have neither explicit mention nor formal definition in the specifications, but that need to be taken into account when developing the generalization algorithms. For example the river that surrounds the area of the toll-plaza is not present in the paper map, without any rule among the specifications explaining why; similarly, two road junctions (the one connecting the toll-plaza to the ordinary roads on the south-east corner of the map and that on the north-west) are less simplified then the other road junctions, with no apparent reason.

As the map can be useful to setup the generalization process, it is usually not the case that there exists a paper map representing the same area that should be generalized; if a map exists, it might be old, making then difficult to know what is missing data and what is generalized data. Furthermore, as the map has to comply with many (cartographic) constraints, it is hard to track back the exact reason behind a certain generalization choice: without formal specifications it might be difficult to tell whether a particular change in the data happened for a precise reason or just for the whim of the cartographer. Hence, particular care should be taken when assessing a generalized result using a map, as not all the discrepancies are caused by errors in the process.

In general, the approach that we chose and the process that we designed showed to be able to handle the problem of generalization and the algorithms that we developed so far produce promising results, although they all show that further research and improvements are needed; as a final remark, the images above depict only six feature classes -although comprising the most important ones- among the 10 times more counted by each data model: as the development of a complete generalization process requires to deal with all of them, this can give a figure of how challenging this endeavor is and how much work is still left to do.

Chapter 8

Conclusions and future work

The research question behind this thesis was whether it was possible to set up an automated generalization process to generalize the Italian medium scale geographical database DB25 in scale 1:25000 from the Regional large scale geographical database in scale 1:5000 GeoDBR.

The results obtained in these three years of research inside the CARGEN project suggest a positive answer to this question: as it was shown in this thesis, the very advanced status of the research on cartographic generalization allowed to design and build a process that met all the requirements

The answer is not definitive yet: some parts of the process are still under development, and the whole process is undergoing continuous refinement and improvements. Furthermore, there still is the pending question of an official evaluation and acknowledgement of the results obtained, that should be carefully examined by professional cartographers. The far reach goal of the CARGEN project has not been reached yet: to modernize the Italian map making process will require further research, time and resources.

This research brought new insights in the field of cartographic generalization in Italy:

the exhaustive analysis of both the GeoDBR and DB25 data models revealed flaws and compatibility issues in both models; this information could be of high interest in the definition of the National geographical database model at 1:5000/1:10000 scale and further on in the definition of the National geographical database models for the 1:25000 and 1:50000 scales;

testing the algorithms on real data, showed on one side the superior quality of a geographical database (the GeoDBR) over the previous topographic base (the CTRN), but on the other revealed that the production processes of the new geographical database are still to be refined and tuned as the sample dataset contained errors and discrepancies with the model;

the design and development of the a whole generalization process brought a gain in the understanding of how this could be done in a country where the research on this field was lacking; furthermore the CARGEN project is the first (and only to time) to investigate the use of the new generation of Italian geographical data, gathering precious first hand information.

This thesis brought also some original contributions to the research field of cartographic generalization:

the algorithm for the generalization of road junctions tackles the problem from a completely new point of view, producing, in our opinion, a drastic improvement over previous approaches, especially when working with large scale data;

the pruning of branches of the hydrography network on the basis of the density is an approach that we could not find any previous mention of in literature; furthermore this approach is scale independent and could be adapted to other type of data (e.g. roads) and it can be then inserted among the set of general algorithms for network pruning;

the typification of ditches is approached in a strictly geometric fashion, modeling it as a problem of even distribution in a bounded space; the simple algorithm developed returns robust and consistent results providing a solution to a topic not covered in literature; the same geometric approach has been also extended to a higher dimensionality in the "matrix" typification of silos, showing the extensibility of this method to the typification of other objects;

the square-ability test developed avoids to apply the squaring algorithm to features that will suffer a big distortion because of it; this test, the first in literature to our knowledge, allows more freedom in applying the squaring algorithm, thus gaining an improvement on the quality of the data, especially at large scales;

in general, the development of a generalization process for the large-medium scales is something rather new, as the works found in literature usually aim at a different scale range; despite the approach chosen for the project was rather pragmatic and thus did not set any particular theoretic background, I believe that the many concrete results achieved, both big or small, may be useful to others attempting such a challenge.

The results obtained, finally, indicate a direction for further developments.

A more dynamic process

The process developed is a big batch process comprised of almost black-box steps executed sequentially; this choice was taken to provide modularity in the initial development stage of the project, when it was not clear how each generalization algorithm would be shaped. In the future the process could gain in performance becoming more dynamic, that is allowing communication among the different generalization algorithms and, in general, interaction among them.

Global quality evaluation step

At present the process lacks a global evaluation step able to measure the quality of the data produced: every algorithm is responsible for the quality of the results it returns and, if instructed to do so, it will try to harmonize its output with the rest of the data; nevertheless a step that performs a quality validation of the data is necessary to move the project one step further. This step could evaluate the data at the end of the process, or provide methods to evaluate it during the execution; the results of such an evaluation could then be used either to trigger a new generalization or to communicate with the user, eventually starting an interactive editing session.

Moving to bigger datasets: intelligent partitioning

One of the future challenges of the project will be to process bigger and bigger datasets; this will require processing the data in batches, i.e. to divide it in parts and process each of them individually. As the global result should not be affected by how the input dataset has been divided, it will be necessary to devise how to tile the data in a way that is transparent to the generalization process (intelligent partitioning). Although this aim might seem dazzling considered the holistic nature of generalization, it is my belief that a point that could be exploited is the intrinsic “locality” of the algorithms: in fact each algorithm can not see “everything” as the holistic notion would require (actually it would require to see even more than what there is), rather its implementation implicitly bounds its actions to a range; this range is actually twofold: an “effect range” (how far the effect of the algorithm could be “felt”) and an “awareness range” (how far the algorithm sees to get informed on how to operate). These ranges may change from algorithm to algorithm, from feature class to feature class, but are known to the developers and, if formalized in some way (as for instance through an interface), could let a partition algorithm to divide the dataset in such a way to not affect (or minimizing the effect) the generalization performed by each algorithm.

From DLM to DCM

Another challenge for the future will be to change the output of the process, generalizing not only the digital landscape model, but also the digital cartographic model, in order to be able to print the generalized data as a real map. This will require to deal with topics that have not been in the main focus of the research, as displacement, symbolization and label placement.

From 1:25000 to 1:50000

Finally, as the aim of the CARGEN project has been extended to the generalization of data at the 1:50000 scale, all the process developed, the understanding and experience gained, the algorithms implemented and everything else should be focused on this new target. While probably some of the work done could be re-used, other will surely need some modification to be adapted and some other should be trashed and will require to start again from scratch. For sure in venturing into this new endeavor it will be possible to count on the passion and the skillful research companions that helped me to arrive till here.

List of publications

De Gennaro M., Rumor M., Savino S., Le procedure per la derivazione del DB25 dal DBT della Regione del Veneto: risultati del progetto CARGEN, Bollettino della Associazione Italiana di Cartografia, 135, Aprile 2009

Savino S., Rumor M., Lissandron I., 2009, Road junction generalization in large scale geodatabases. In A. Krek, M. Rumor, S. Zlatanova, E. M. Fendel (Eds), UDMS Annual 2009, Taylor & Francis

Savino S., Rumor M., Zanon M., Lissandron I., 2010, Data enrichment for road generalization through analysis of morphology in the CARGEN project, ICA Workshop, 2010, Zurich

Savino S., Rumor M., Zanon M., 2011, Pattern Recognition and typification of ditches, 25th Conference of the International Cartographic Association, 2011 Paris (accepted)

Savino S., Rumor M., Canton F., Langiù G., Raineri M., 2011, Model and cartographic generalization of the hydrography network in the CARGEN project, 25th Conference of the International Cartographic Association, 2011 Paris (accepted)

Bibliography

AGENT (2000) Project ESPRIT AGENT Map Generalisation by Multi-Agent Technology online: http://agent.ign.fr

AdV Arbeitsgruppe ATKIS (1989) Amtliches Topographisch - Kartographisches Informationssystem (ATKIS), Gesamtdokumentation, Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder (AdV) ISPRS Working Group IV/3, Hannover

Ai T., van Oosterom P. (2002) GAP-Tree Extension based on skeletons, in Proceedings of ISPRS Symposium on Geospatial Theory, Processing and Application, Ottawa

Aichholzer O., Aurenhammer F. (1995) A novel type of skeleton for Polygons, Journal of Universal Computer Science, 1 (12), 752-761

Anders K.-H., Sester M. (2000) Parameter-free Cluster Detection In Spatial Databases And Its Application To Typification, in: International Archives Of Photogrammetry And Remote Sensing, Amsterdam, Netherlands, 33

Anders K.-H. (2006) Grid Typification, in 12th International Symposium On Spatial Data Handling, Springer-verlag, 633-642

Anders K.-H (2005) Level Of Detail Generation Of 3D Building Groups By Aggregation And Typification, in Proceedings Of 22th International Cartographic Conference 2005, A Coruna, Spain.

Andrea Forberg and Helmut Mayer (2003) Squaring and Scale-Space Based Generalization of 3D Building Data ICA, commission on map generalisation, Paris, France, 28-30 April

Bader M., Barrault M. (2001) Cartographic Displacement in Generalization: Introducing Elastic Beams ICA Workshop on Progress in automated map generalization, Beijing, China, 2-4 August

Bader M., Weibel R. (1997) Detecting and resolving size and proximity conflicts in the generalization of polygonal maps, in Proceedings of the 18th International Cartographic Conference, Stockholm, 1525-1532

Bader, M. (2001) Energy Minimizing Methods for Feature Displacement in Map Generalization PHD thesis, University of Hannover

Baella B., Pla M. (1999) Map generalization to obtain the topographic map of catalonia ICA Workshop on Progress in Automated Map Generalisation, Ottawa, Canada, 12-14 August

Baella B., Pla M. (2000) Some generalization practices on relief representation derived from the Topographic database of Catalonia at scale 1:5 000 in: Buchroithner M. (Editor), High Mountain Cartography 2000. Dresden: Institute for Cartography, Dresden University of Technology and DGFK 7-17

Baella B., Pla M. (2003) An example of database generalization workflow: the Topographic Database of Catalonia at 1:25.000, ICA Workshop on Map Generalization, Paris, 28-30 April

Baella B., Pla M. (2005) Reorganizing the Topographic Databases of the Institut Cartogràfic de Catalunya applying generalization, 8th ICA Workshop on Generalization and Multiple Representation, A Coruña, Spain, 7-8 July

Baella B., Palomar-Vázquez J., Pardo-Pascual J. E., Pla M. (2007) Spot heights generalization: deriving the relief of theTopographic Database of Catalonia at 1:25,000 from the master database, 10th ICA Workshop on generalisation and multiple representation, Moscow, 2-3 August

Balboa J. L. G., Lopez F. J. A. (2009) Sinuosity pattern recognition of road features for segmentation purposes in cartographic generalization, in Pattern Recognition, 42(9), 2150-2159

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

Blanca Baella, Maria Pla (2005) Reorganizing the Topographic Databases of the Institut Cartogràfic de Catalunya applying generalization 8th ICA WORKSHOP on Generalization and Multiple Representation, A Coruña, July 7-8

Boffet A. (2001) Méthode de création d'informations multi-niveaux pour la généralisation cartographique de l'urbain, PhD Thesis, Université de Marne la Vallée, France

Brassel K., Weibel R. (1988) A review and conceptual framework of automated map generalization International Journal of Geographical Information Science, 2 (3), 229-244

Braun A., Halbecq X., Lecordix F., Le Gallic J. -M., Prigent F. (2007) A new flowline for the French topographic maps in I.G.N. Proceedings of the 22nd International Cartographic Conference, Moscow, Russia, 4-10 August

Brewer C. A., Buttenfield B. P., Usery E.L. (2009) Evaluating generalizations of hydrography in differing terrains for The National Map of the United States, in Proceedings of the 24th International Cartographic Congress, Santiago, Chile, 15-21 November

Burghardt D., Cecconi A. (2007) Mesh Simplification For Building Typification in International Journal Of Geographical Information Science, 21(3), 283–298

Burghardt D., Schmid S. (2009) Constraint-Based Evaluation of Automated and Manual Generalised Topographic Maps Cartography in Central and Eastern Europe, Lecture Notes in Geoinformation and Cartography, 2, 147-162

Busch A., Gerke M., Grünreich D., Heipke, Helmholz P., Liedtke C.-E., Müller S. (2006) Automated verification of a topographic dataset using Ikonos imagery Symposium ISPRS - Commission IV, Goa

Buttenfield B. P., Wolf E. (2007) The Road and the River Should Cross at the Bridge Problem: Establishing Internal and Relative Topology in an MRDB ICA Workshop on Map Generalization, Moscow, 2-3 August

Buttenfield B. P., McMaster R. B. (Editors) (1991) Map generalization: making rules for knowledge representation, Longman Proceedings of the 8th International Symposium on Spatial Data Handling, Vancouver, Canada, 12-15 July, Longman, 214-224

Buttenfield, B. P. (1995) Object-Oriented Map Generalization In: GIS and Generalization, Ed JC Müller, JP Lagrange & R. Weibel: Taylor & Francis, 91-105

CARGEN (2009) La generalizzazione del DB25 a partire dal GeoDBR 1:5000 - Schema della derivazione, query per il popolamento e operazioni sulle geometrie, Dipartimento di Ingegneria dell'Informazione, Padova

CNIPA (2009) Centro NAzionale per l'Informatica nella Pubblica Amministrazione Catalogo di dati territoriali – Specifiche di contenuto per i DB Geotopografici, online: http://www.cnipa.gov.it/site/it-IT/Attivit%C3%A0/Sistemi_Informativi_Territoriali/Specifiche_tecniche/DB_Geotopografici/

Cecconi A. (2003) Integration of Cartographic Generalization and Multi-Scale Databases for Enhanced Web Mapping, PhD Thesis, Department of Geography, University of Zurich

Chaudhry O., Mackaness W. A. (2005) Rural and Urban Road Network Generalization Deriving 1:250,000 form OS MasterMap, Proceedings of the 22nd International Cartographic Conference, A Coruña, Spain, 9-16 July

Chin F., Snoeyink J., Wang C. A. (1995) Finding the medial axis of a simple polygon in linear time, Proceeeding of the 6th annual international symposium on algorithms and computation, 382-391

Chithambaram R., Beard K., Barrera R. (1991) Skeletonizing polygons for map generalization, in Technical Papers, ACSM-ASPRS Convention, Cartography and GIS/LIS, 2, 44-55

Christophe S., Ruas A. (2002) Detecting Buildings alignments for generalisation purposes in: Richardson D. E., Peter van Oosterom P. v. (Editors) Advances in spatial data handling: 10th International Symposium on Spatial Data Handling, 1, Springer 420-432

DGIWG (2000) The Digital Geographic Information Exchange Standard (DIGEST): Feature and Attribute coding catalogue Digital Geographic Information Working Group

Djouadi, Y. (1994) Cartage: A cartographic layout system based on genetic algorithms In: Proceedings of the Fifth European Conference and Exhibition on Geographical Information Systems 48–56

Douglas D., Peucker T. (1973) Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, in Cartographica: The International Journal for Geographic Information and Geovisualization, 10, 112-122

Duchêne C., Ruas A. (2001) Généralisation de données géographiques. Présentation des résultats du projet AGENT Bulletin d'information de l'IGN, n°72 : Bilan de la Recherche

Duchêne C., Bard S., Barillot X., Ruas A., Trévisan J., and Holzapfel F. (2003) Quantitative and qualitative description of building orientation 5th ICA Workshop on progress in automated map generalization, Paris, France, 28-30 April

Dutton G. (1999) Scale, Sinuosity, and Point Selection in Digital Line Generalization, in Cartography and Geographic Information Science, 26(1), 33-53

ESRI (1996) Automation of Map Generalization, The cutting-Edge Technology ESRI White Paper Series

ESRI (2004) What is ArcGIS ESRI Press, ISBN 1-58948-090-2

ESRI (2008) ArcObjects JavaDOC online: http://resources.esri.com/help/9.3/arcgisengine/java/api/arcobjects/index.html

Edwardes A.J., Mackaness W. A. (2000) Intelligent Generalization of Urban Road Networks proceedings of GIS Research UK Conference (GISRUK'2000), York, UK, 81-85

Eppstein D., Erickson J. (1999) Raising Roofs, Crashing Cycles, and Playing Pool: Applications of a Data Structure for Finding Pairwise Interactions, Discrete Computational Geometry, 22, 569-592

Fan H., Meng L. (2010) A generic approach for simplification of buiding ground plan 13th ICA Workshop on Generalisation and Multiple Representation, Zurich, 12-13 September

Féchir A., De Waele J. (2006) The future production of generalised maps at IGN Belgium Multiple representation and interoperability of spatial data ISPRS Workshop, Hannover, Germany, 22-24February

Foerster T., Stoter J., Kraak M.-J. (2010) Challenges for automated generalisation at European Mapping Agencies: a qualitative and quantitative analysis The Cartographic Journal, 47 (1), 41-54

Frank Brazile (2000) Semantic Infrastructure and Methods to Support Quality Evaluation in Cartographic Generalization PhD Thesis, Department of Geography, University of Zurich

Frank, R., Ester, M. (2005) Evaluating the quality of generalized maps by using a quantitative map similarity measure 13th ACM International Symposium of GIS, Bremen, Germany, 12-13 November

GEMURE (2005) The GEMURE project. Multi-scale generalization and multiple representation techniques for efficient production and delivery of cartographic data Final Report ESRI White Paper Series

Gaffuri J. (2007) Outflow preservation of the hydrographic network on the relief in map generalisation, International Cartographic Conference, Moscow, Russia, 4-9 August

Galanda, M., Weibel, R (2002) An Agent-Based Framework for Polygonal Subdivision Generalization In: Richardson D. and van Oosterom P.(eds.): Advances in Spatial Data Handling. 10th International Symposium on Spatial Data Handling, Springer-Verlag, 121-136

Garcia J. L., Ariza F. J. (2000) Frequency Filtering of Linear Features by Means of Wavelets. A method and an Example, The Cartographic Journal, 37(1), 39-49

GeoTools (2002) GeoTools, The Open Source Java GIS Toolkit online: http://www.geotools.org/

GiMoDig project (2001) Geospatial info-mobility service by real-time data-integration and generalisation, online: http://gimodig.fgi.fi/index.php

Gokgoz T. (2005) Generalization of contours using deviation angles and error bands, The Cartographic Journal, 42 (2), 145-156

Gosling J., McGilton H. (1996) The Java Language Environment online: http://www.oracle.com/technetwork/java/langenv-140151.html

Gruenreich D. (1985) Computer assisted generalisation, Papers CERCO-Cartography course, Institut fur Angenwandte Geodaesie, Franfkur a.M.

Gruenreich D. (1995) Development of computer-assisted generalization on the basis of cartographic model theory, in: Mueller J. C., Lagrange J. P., Weibel R. (Editors), GIS and generalization: methodology and practice, Taylor&Francis, 47-55

Hampe M., Sester M. (2004) Generating and Using A Multi-Representation Data-Base (MRDB) for Mobile Applications, 7th ICA Workshop on Generalisation and Multiple Representation, Leicester, United Kingdom, 20-21 August

Harrie L., Sarjakoski L. T., L. Lehto (2002) A variable-scale map for small-display cartography, Proceedings of the Joint International Symposium on “GeoSpatial Theory, Processing and Applications” (ISPRS/Commission IV, SDH2002), Ottawa, Canada, 8-12 July

Harrie L., Weibel R. (2007) Modelling the overall process of generalization in: Mackaness W. A., Anne Ruas A., Sarjakoski L. T. (Editors), Generalisation of geographic information: Cartographic modelling and applications, Elsevier Science, 67-87

Haunert J. H., Sester M. (2004) Using the Straight Skeleton for Generalisation in a Multiple Representation Environment, ICA Workshop on Generalisation and Multiple Representation, Leicester, United Kingdom, 20-21 August

Haunert J. H., Sester M. (2008) Area Collapse and Road Centerlines based on Straight Skeletons, in: GeoInformatica 12 (2), 169-191

Haunert J.H., Wolf A. (2008) Optimal simplification of building ground plans International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science, 37 (B2)

Heinzle F., Anders K.-H., Sester M. (2005) Graph Based Approaches for Recognition of Patterns and Implicit Information in Road Networks, Proceedings of the 22nd International Cartographic Conference, A Coruña, Spain, 9-16 July

Helmholz P, Ziems M (2000) System development for automated quality control and update of geo-data http://www.ipi.uni-hannover.de/html/forschung/laufend/wipka/wipka.htm

Hershberger J., Snoeyink J. (1992) Speeding Up the Douglas-Peucker Line-Simplification Algorithm, in Proceedings of the 5th International Symposium on Spatial Data Handling, 134-143

Hesse, W., Leahy, F. (1990) The Authoritative Topographic Cartographic Information System ATKIS Consultancy Report for Intergraph Federal Systems

Hoppe H., DeRose T., Duchamp T., McDonald J, Stuetzle W. (1993) Mesh Optimization Computer Graphics, 27 (Annual Conference Series), 19-26

Hoppe H. (1996) Progressive Meshes, Computer Graphics, 30 (Annual Conference Series), 99-108

Horton R. E. (1945) Erosional Development of Streams and Their Drainage Basins; Hydrophysical Approach to Quantitative Morphology. in Bulletin of the Geological Society of America, 56, 275-370

IGMI (2006) Struttura del DB25: feature attributi e domini Istituto Geografico Militare Italiano, 2a Direzione della Produzione e Direzione Lavori Ricerca e Sviluppo, Firenze

Intesa GIS / WG 01 (2004) Specifiche di contenuto: gli strati, i temi e le classi In: specifiche per la realizzazione dei database topografici di interesse generale

Istituto Geografico Militare (2006) Struttura del DB25: feature attributi e domini Seconda direzione della produzione e direzione lavori di ricerca e sviluppo.

JTS (2002) Java Topology Suite, Vivid Solutions online: http://tsusiatsoftware.net/jts/main.html

Jabeur, N., Moulin, B. (2004) A multiagent based approach for the resolution of spatial conflicts European Workshop on Multi-Agent Systems (EUMAS), Barcelona, 333-343

Jahard Y., Lemarié C., Lecordix F. (2003) The implementation of new technology to automate map generalisation and incremental updating processes Proceedings of the 21st International Cartographic Conference, Durban, South Africa, 10-16 August, 1449-1459, CD-ROM

Jiang B., Claramunt C. (2002) A Structural Approach to Model Generalisation of an Urban Street Network, 5th AGILE Conference on Geographic Information Science, Palma, Spain, 25-27 April

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

Joao E. M. (1995) Importance of quantifying the effects of generalization in: Mueller J. C., Lagrange J. P., Weibel R. (Editors), GIS and generalization: methodology and practice, Taylor&Francis, 183-193

Jones C., Bundy G., Ware M. (1995) Map generalisation with a triangulated data structure, Cartography and Geographic Information Systems, 22 (1), 55-69

Karam R., Favetta F., Laurini R., Chamoun R. (2010) Integration of Similar Location Based Services Proposed by Several Providers (In NDT 2010: Network and Digital Technologies, LNCS Springer ed. Prague. pp. 1-9. LNCS Tome . Springer . 2010. See pdf.) Designing Adaptable Geographic Objects for Mobile Applications In Proceedings of the 4th International Conference on Web Information Systems Engineering (WISE 2003), Roma, 13 December, 99-152

Katalin Toth (2007) Data consistency and multiple-representation in the ESDI ICA Workshop on Map Generalization, Moscow, 2-3 August

Kilpelainen T., Sarjakosky T. (1995) Incremental generalization for multiple representations of geographical objects in: Mueller J. C., Lagrange J. P., Weibel R. (Editors), GIS and generalization: methodology and practice, Taylor&Francis, 209-218

Kilpeläinen, T. (2000) Maintenance of Multiple Representation Databases for Topographic Data The Cartographic Journal Vol. 37, No 2, 101-107

Kraak, M.J., Ormeling, F. J., (2003) Cartography: visualization of geospatial data, 2 edition. Harlow, England: Prentice Hall

Label-EZ (2005) MapText Inc. online: http://www.maptext.com/

Lamy, S., Ruas, A., Demazeau, Y., Jackson, M., Mackaness, W., Weibel, R. (1999) The application of Agents in Automated Map Generalisation Proceedings of 19th ICA meeting. Ottawa, Canada, 12-14 August

Lecordix F., Jahard Y., Lemarié C., Hauboin E. (2005) The end of Carto2001 Project TOP100 based on BD Carto® database, 8th ICA Workshop on generalisation and multiple representation, A Coruña, Spain, 7-8 July

Lecordix F., Le Gallic J. -M., Gondol L. (2007) Development of a new generalisation flow line for topographic maps 10th ICA Workshop on Map Generalization, Moscow, Russia, 2-3 August

Lecordix F., Lemarié C. (2007) Managing generalisation updates in IGN map production in: Mackaness W. A., Anne Ruas A., Sarjakoski L. T. (Editors), Generalisation of geographic information: Cartographic modelling and applications, Elsevier Science, 285-300

Lecordix F., Trévisan J., Le Gallic J. -M., Gondol L. (2006) Clarity experimentations for cartographic generalisation in production. 9th ICA Workshop on Generalisation and Multiple Representation, Portland, USA, 25 June

Lee D. T. (1982) Medial Axis Transformation of a Planar Shape, IEEE Transactions on pattern analysis and machine intelligence, PAMI-4(4), 363-369

Lemarié C. (2003), Generalisation process for Top100: research in generalisation brought to fruition, ICA Workshop on Map Generalization, Paris, 28-30 April

Li Z., Yan H., Ai T., Chen J. (2004) Automated building generalization based on urban morphology and Gestalt theory International Journal of Geographical Information Science, 18 (5), 513-534

Mostaccio C., Laurini R., Gordillo S., Bazzocco J., Rossi R. (2003) Designing Adaptable Geographic Objects for Mobile Applications In Proceedings of the 4th International Conference on Web Information Systems Engineering (WISE 2003), Roma, December 13th, 2003.

Mackaness W. A., Mackechnie G. (1999) Detection and Simplification of Road Junctions in Automated Map Generalization, pp. in GeoInformatica, 3, 185-200

Mackaness W. A., Perikleous S., Chaudhry O. Z. (2008) Representing Forested Regions at Small Scales: Automatic Derivation from Very Large Scale Data, The Cartographic Journal, 45 (1), 6-17

Mackaness W. A. (1994) An algorithm for conflict identification and feature displacement in automated map generalization Cartography and Geographic Information Systems, 21 (4), 219-232

Mackaness W. A. (1995) Analysis of Urban Road Networks to Support Cartographic Generalization, in Cartograpy and Geographic Information Systems, 22, 306-316

Mackaness, W. A., Beard M. K. (1993) Use of Graph Theory to Support Map Generalization, in Cartography and Geographic Information Systems, 20, 210-221

Mackaness, W.A. (1997) Design as process - strategies for automated map design, Position paper for the 2nd workshop on progress in automated map generalization, Gävle, Sweden.

Mackanness W. A., Ruas A. (2007) Evaluation of the Map Generalization Process, Generalisation of geographic information: Cartographic modelling and applications, William A. Mackaness, Anne Ruas, L. Tiina Sarjakoski (Editors), Elsevier Science; 1st edition (June 6, 2007), 89-111

McMaster R. B., Shea K. S. (1988) Cartographic generalization in a a Digital Environmnent: a framework for implementation in a gegographic information system GIS/LIS'88 Proceedings, San Antonio, Texas, November 30-December 2, 240-249

McMaster R. B., Shea K. S. (1992) Generalization in Digital Cartography Association of American Geographers, Washington D.C

McMaster R. B. (1987) Automated line generalization, Cartographica, 24 (2), 74-111

McMaster R. B. (2003) The Creation of Multiple Scale Databases in the NHGIS ICA Workshop on Map Generalization, Paris, 28-30 August

Meng L. (1997) Automatic Generalization of Geographic Data Technical Report, VBB Viak, Stockholm, Schweden, online: http://www.vbbviak.sweco.se/Research_net/preport/fm9706.htm

Mueller J. C. (1990) The removal of spatial conflicts in line generalization Cartography and Geographic Information Systems, 17 (2), 141-149

Muller J. C., Wang Z. (1992) Area-Patch generalisation: a competitive approach, The Cartographic Journal, 29 (1), 137-144

Müller J. C. (1991) Generalisation of Spatial Databases In Geographical Information Systems, D. J. Maguire, M. Goodchild and D. Rhind, (Editors) London: Longman Scientific

Müller, J. C., Weibel R., Lagrange J. P., Salgé F. (1995) Generalisation: State of the Art and Issues in: Mueller J. C., Lagrange J. P., Weibel R. (Editors), GIS and generalization: methodology and practice, Taylor&Francis, 3-18

Müller, J. C. (1991) Generalisation of Spatial Databases In Geographical Information Systems, D. J. Maguire, M. Goodchild and D. Rhind, (ed) London: Longman pp. Scientific

Neun M., Weibel R., Burghardt D. (2004) Data enrichment for adaptive generalisation 7th ICA Workshop on generalisation and multiple representation, United Kingdom, 20-21 August

Novit Kreiter (2003) Multirepresentation Databases and Need for Generalization at Swisstopo ICA Workshop on Map Generalization, Paris, 28-30 April

OpenJUMP (2004) Jump Pilot Project online: http://www.openjump.org/

Oracle (2005) Spatial Java API Reference, Online: http://download.oracle.com/docs/cd/B19306_01/appdev.102/b14373/toc.htm

Oracle (2005) Spatial User's Guide and Reference online: http://www.oracle.com/pls/db102/to_pdf?pathname=appdev.102%2Fb14255.pdf&remark=portal+%28Books%29

Painho M. (1995) Effects of generalization on attribute accuracy in: Mueller J. C., Lagrange J. P., Weibel R. (Editors), GIS and generalization: methodology and practice, Taylor&Francis, 194-206

Pantazis D., Donnay J.-P, Billen R. (2006) GIS design and implementation methjods: The MECOSIG improvement and updating project 25th Urban Data Management Symposium, UDMS, Aalborg, May 15-17

Pantazis D. N., Mpatsos E. , Poliths P., Kassolis M., Yingcheng L., Guo T.(2007) Design of multiscale cartographic databases: The Greek Case In proceedings of the 26th Urban and Regional Data Management Symposium, Stuttgart

Pantazis D. (1994) La mèthode de conception de S.I.G. ME.CO.S.I.G. et le formalisme CON.G.O.O. (CONception Gèographique Orientèe Objet) EGIS/MARI Proceedings, Paris

Parent C., Spaccapietra S., Zimanyi E. (2000) MurMur: Database Management of Multiple Representations AAAI-2000 Workshop on Spatial and Temporal Granularity, Austin, Texas, July 30

Pavelsky T. M., Smith L. C. (2008) RivWidth: A Software Tool for the Calculation of River Widths From Remotely Sensed Imagery, in IEEE Geoscience and Remote Sensing Letters, 5 (1)

Pilon D., Beaulieu A., Sabo N. (2010) The Generalization of the Canadian Landmass: A Federal Perspective, in 13th Workshop of the ICA commission on Generalisation and Multiple Representation, Zurich, 12-13 September

Pla, M., Sánchez, S., Colomer, J. L. (2003) Updating topographic databases: the example of the Topographic Database of Catalonia at scale 1:5 000 Presentat a la Updating and Versioning Commission de la ICA

Plazanet C. (1995) Measurements, characterisation and classification for automated linear features generalisation, in Proceedings AutoCarto 12, Charlotte, 4, 59-68

Prasaad L. (1997) Morphological Analysis of Shapes CNLS Newsletter 139 (1), 1-18

Rainsford D., Mackaness W. (2002) Template Matching in Support of Generalisation of Rural Buildings In: Richardson D., Oosterom P.v. (Editors), Advances in Spatial Data Handling, Springer, 137-151.

Ratajski L. (1967) Phénomènes des points de généralisation International yearbook of cartography, 7, 143-151

Refractions Research, (2001) PostGIS online: http://postgis.refractions.net/

Refractions Research, (2010) PostGIS JDBC JavaDOC, online: http://postgis.refractions.net/documentation/javadoc

Regione Veneto (2004) Carta Tecnica Regionale Numerica: codifiche Segreteria Regionale per il Territorio, Unità complessa per il Sistema Informativo Territoriale e la Cartografia

Regione Veneto (1992) Carta Tecnica Regionale Numerica - Codifiche Regione del Veneto, Segreteria regionale per il territorio, Unità complessa per il Sistema Informativo Territoriale e la Cartografia. online: http://www.regione.veneto.it/NR/rdonlyres/26AFDADC-13AE-4584-9BA4-3FF395CE2A22/0/Codifiche5000e10000.pdf

Regione Veneto (2008) DB Topografico Segreteria Regionale all'Ambiente e Territorio, Unità di Progetto per il SIT e la cartografia. online: http://www.regione.veneto.it/NR/rdonlyres/50B13921-AD61-41A0-98CD-9429940E7B38/0/DBTopografico26.pdf

Regnauld N., McMaster R. B. (2007) A Synoptic view of generalisation operators in: Mackaness W. A., Anne Ruas A., Sarjakoski L. T. (Editors), Generalisation of geographic information: Cartographic modelling and applications, Elsevier Science, 37-66

Regnauld N. (2001) Contextual Building Typification In Automated Map Generalization in Algorithmica, 30, 312-333

Regnauld N. (2003) Algorithms for the amalgamation of topographic data, Proceedings of the 21st International Cartographic Conference, Durban, South Africa, 10-16 August

Regnault N. (1996) Recognition Of Building Clusters For Generalization, Proceedings of the 7th International Symposium on Spatial Data Handling, Delft, The Netherlands, 185-198

Renard J., Gaffuri J., Duchene C. (2010) Capitalisation problem in research - example of a new platform for generalisation: CartAGen 2010, 13th ICA Workshop on Generalisation and Multiple Representation, Zurich, 12-13 September

Revell P., Regnauld N., Thom S. (2005) Generalising OS MasterMap topographic Buildings and ITN road centrelines to 1:50 000 scale using a spatial hierarchy of agents, triangulation and topology Proceedings of the 22nd International Cartographic Conference, A Coruña, Spain, 9-16 July

Revell P., Regnauld N., Thom S. (2006) Generalising and Symbolising Ordnance Survey Base Scale Data to create a prototype 1:50000 scale vector map 9th ICA Workshop on Generalisation and Multiple Representation, Portland, USA, 25 June

Revell P. (2005) Building on Past Achievements: Generalising OS MasterMap® Rural Buildings to 1:50 000 7th ICA Workshop on Generalisation and Multiple Representation, Leicester, United Kingdom, August 20-21

Revell P. (2005) Seeing the Wood from the Trees: Generalising OS MasterMap Tree Coveage Polygons to Woodland at 1:50000 Scale, 8th ICA Workshop on Generalisation and Multiple Representation, A Coruña, Spain, July 7-8

Revell P. (2010) The prototype of the VectorMap District serie (oral presentation) 13th ICA Workshop on Generalisation and Multiple Representation, Zurich, 12-13 September

Rieger, M., Coulson, M. (1993) Consensus or Confusion: Cartographer's Knowledge of Generalization Cartographica, 30 (2&3), 69-80

Ruas A, Plazanet C. (1997) Strategies for automated generalization In: Kraak M. J., Molenaar M., Fendel E. M. (Editors), Advances in GIS Research II, Proceedings of the 7th International Symposium on Spatial Data Handling, Delft, The Netherlands, Taylor & Francis, 319-336

Ruas A. (2000) Project AGENT: Overview and Results of a European R&D Project in Map Generalisation ICA Workshop, Barcelona, Spain, 21-23 November

Ruas A. (2001) Automatic Generalisation Project: learning process from interactive generalisation, OEEPE, European ORganization for Experimental Photogrammetric Research, Official Publication (39), Frankfurt a.M.:Bundesamt fur Kartographie un Geodaesie

Ruas, A, Plazanet, C. (1996) Strategies for automated generalization In: Proceedings of the 7th International Symposium on Spatial Data Handling, Advances in GIS Research II, 1996, 6.1-6.18

Ruas, A., Lagrange, J.P. (1995) Data and knowledge modelling for generalisation, in: Müller J. -C., Lagrange J. -P., Weibel R. (Editors) GIS and generalization, Methodology and Practice, GISDATA 1, Taylor & Francis, 73-90

Ruas, A. (1998) A Method for building displacement in automated map generalisation, International Journal on GIS, 1998, 2 (8), 789-803

Rusak M. E., Castner H. W. (1990) Horton's ordering scheme and the generalisation of river networks, in The Cartographic Journal, 27, 104-112

Rütschlin J. (2000) The requirements for a component-based architecture, Proceedings of the ProSTEP Science Days 2000 "SMART Engineering", Stuttgart, Germany, 13-14 September, 252-260

Servigne S., Ubeda T., Puricelli A., Laurini R. (2000) A Methodology for Spatial Consistency Improvement of Geographic Databases, GeoInformatica, 4 (1) 7-34

Saalfeld A. (1999) Topologically Consistent Line Simplification with the Douglas-Peucker Algorithm, in Cartography and Geographic Information Science, 26 (12), 7-18

Sarjakosky T. (2007) Conceptual models of generalisation and multiple representation in: Mackaness W. A., Anne Ruas A., Sarjakoski L. T. (Editors), Generalisation of geographic information: Cartographic modelling and applications, Elsevier Science, 11-37

Sester M., Brenner C. (2000) Typification Based On Kohonen Feature Nets Proceedings Of The 1st International Conference On Giscience, Savannah, 21–22

Sester M. (2000) Generalization Based on Least Squares Adjustment In: ISPRS (ed.) International Archives of Photogrammetry and Remote Sensing.

Shea K. S., McMaster R. B. (1989) Cartographic generalization in a a Digital Environmnent: when and how to generalize AUTO-CARTO, 9th International Symposium on Computer Assisted Cartography, Baltimore, Maryland, April 2-7, 56-67

Sheehan R. (2001) Generalisation at the Danish Mapping Agency - A Solution Based on the AGENT Research Project and Laser-Scan Object-oriented Technology pp. GIM International 15(7)

Simav O., Aslan S., Çetinkaya B., Çobankaya O. N. (2010) Implementation of Comprehensive Modeling Techniques on KARTOGEN Generalization Software 13th ICA Workshop on Generalisation and Multiple Representation, Zurich, 12-13 September

Smith V. (2003) Generalization for Medium-Scale Mapping: Results and Statistics from One Production Implementation, ICA Workshop on Map Generalization, Paris, 28-30 April

Stanislawski L., Barbara Buttenfield B., Samaranayake V. A. (2010) Generalization of Hydrographic Features and Automated Metric Assessment Through Bootstrapping in 13th Workshop of the ICA commission on Generalisation and Multiple Representation, Zurich, Switzerland, 12-13 September

Stanislawski L. (2009) Feature Pruning by Upstream Drainage Area to Support Automated Generalization of the United States National Hydrography Dataset, in Computer, Environment and Urban Systems, 33, 325-333

Stansilawski L.V., Buttenfield B.P., Finn, M.P., Roth K. (2009) Stratified Database Pruning to Support Local Density Variations in Automated Generalization of the United States National Hydrography Dataset, in Proceedings of 24th International Cartographic Conference, Santiago, Chile

Steiniger S., Bocher E. (2009) An Overview on Current Free and Open Source Desktop GIS Developments International Journal of Geographical Information Science, 23 (10), 1345-1370

Stoter J. (2005) Generalisation within NMA's in the 21st century Proceedings of the 22nd International Cartographic Conference, A Coruña, Spain, 9-16 July

Stoter J. (2005) Generalisation: the gap between research and practice, 8th ICA Workshop on generalisation and multiple representation, A Coruña, Spain, 7-8 July

Stoter J. (2007) EuroSDR project, Research on state-of-the-art of generalisation. Oral presentation during the Workshop of the ICA Commission on Generalisation and Multiple Representation, Moscow, Russia, 2-3 August, online http://aci.ign.fr/Moscow/oral/jantien.pdf

Stoter J. (2010) State-of-the-Art of Automated Generalisation in Commercial Software, Report of EuroSDR projects, online: http://www.eurosdr.net/publications/58.pdf

Strahler A. N. (1952) Hypsometric (Area-altitude) Analysis Of Erosional Topography, in Geological Society Of America Bulletin, 63 (11), 1117-1142

The MurMur Consortium (2000) Supporting Multiple Representations in Spatio-Temporal databases 6th EC-GI & GIS Workshop, Lyon, France, June 28-30

Thom S. (2005) A Strategy For Collapsing Os Integrated Transport Network Dual Carriageways 8th Ica Workshop On Generalisation And Multiple Representation, A Coruña, Spain, July 7-8

Thomson R. C, Richardson D. E. (1995) A Graph Theory Approach to Road Network Generalization, in Proceedings of the 17th ICA Meeting, Barcelona, Spain, 1871-1880

Thomson R. C, Richardson D. E. (1999) The 'Good Continuity' Principle of Perceptual Organisation applied to the Generalisation of Road Networks, in Proceeding of the 19th International Cartographic Conference, Ottwawa, 1215-1225

Thomson R. C., Brooks R. (2002) Exploiting perceptual grouping for map analysis, understanding and generalization: The case of road and river networks, in Graphics Recognition, Algorithms and Applications, 148–157

Thomson R. C., Brooks R. (2000) Efficient generalization and abstraction of network data using perceptual grouping, 5th International Conference on GeoComputation, Medway, United Kingdom, 23-25 August, http://www.geocomputation.org/2000/GC029/Gc029.htm

Thomson R. C., Brooks R. (2007) Generalization of Geographical Networks, in: Generalisation of geographic information: Cartographic modelling and applications, William A. Mackaness, Anne Ruas, L. Tiina Sarjakoski (Editors), Elsevier Science; 1st edition (June 6, 2007), 255-267

Tinghua A., Yaolin L., Chen J. (2006) The Hierarchical Watershed Partitioning and Data Simplification of River Network in Progress in Spatial Data Handling, Part 11, 617-632

Topfer F., Pillewizer W. (1966) The principles of selection The Cartographic Journal, 3 (1), 10-16

Touya G., Duchêne C., Ruas A. (2010) Collaborative generalisation: Formalisation of generalisation knowledge to orchestrate different cartographic generalisation processes, 6th international conference on Geographic Information Science, Zurich, Switzerland, 14-17 September

Touya G. (2007) A Road Network Selection Process Based on Data Enrichment and Structure Detection, Proceedings of the 10th ICA Workshop on generalisation and multiple representation, Moscow, Russia, 2-3 August

Touya G. (2007) River Network Selection based on Structure and Pattern Recognition Proceedings of International Cartographic Conference, Moscow, Russia, 4-9 August

Touya G. (2008) First thoughts for the orchestration of generalisation methods on heterogeneous landscapes 11th ICA Workshop on generalisation and multiple representation, Montpellier, France, 20-21 June

Turk G. (1992) Re-Tiling Polygonal Surfaces, Computer Graphics, 26 (2), 55-64

Töpfer F., Pillewizer W. (1966) The principles of selection: a means of cartographic generalization, in The Cartographic Journal 3 (1), 10-16

Van Kreveld M., Peschier J. (1998) On the Automated Generalization of Road Network Maps, proceedings of the 3rd International Conference on Geocomputation, Bristol, United Kingdom

Van Oosterom P, Schenkalaars V. (1993) The design and implementation of multi-scale GIS Proceedings of EGIS'93, Italy

Van Oosterom P, Schenkalaars V. (1996) Applying reactive data structure in an interactive multi-scale GIS Molenaar & Martien Edition. Methods for the generalization of geo-databases, Netherlands geodetic commission, 1996. No. 43

Van Oosterom, P. (1995) The GAP-tree, an approach to 'on-the-fly' map generalisation of an area partitioning, In: GIS and Generalization: Methodology and Practice, Taylor&Francis, 120-132.

Vangenot C., Parent C., Spaccapietra S., (1999) Multi-representations and multiple resolutions in geographic databases Advanced Database Symposium'99, IPSJ Symposium Series Vol.99, N=B019, Tokyo, Japan, December 6-7

Veepuri S. (2007) Evaluation of ArcGIS and CHANGE PUSH TYPIFY software for the automatic generalization of large scale topographic data. Enschede

Ware J. M., Jones C. B. (1998) Conflict reduction in map generalization using iterative improvement, Geoinformatica, 2 (4), 383-407

Weibel R., Dutton G. (1998) Constraints-based Automated Map Generalization Proceedings of the 8th International Symposium on Spatial Data Handling, Vancouver, Canada, July 12-15, 214-224

Weibel R., Dutton G. (1999) Generalising spatial data and dealign with multiple representations in Longley P. A., Goodchild M. F., Maguire D. J., Rhind D. W. (Editors), Geographical Information Systems, 1: Principles and Techinical Issues, 2nd ed., Wiley & Sons, 125-155

Weibel R., Keller S., Reichenbacher T. (1995) Overcoming the knowledge acquisition bottleneck in map generalization: the role of interactive systems and computational intelligence In: Frank U., Kuhn W. (Editors), Spatial Information Theory, A Theoretical Basis for GIS, COSIT'95, Semmering, Austria, September 21-23, Lecture Notes in Computer Science (988), Springer, 139-156

Weibel R. (1991) Amplified intelligence and rule-based systems In: Buttenfield B. P., McMaster R. B., (Editors) Map generalization: making rules for knowledge representation, Harlow: Longman group, 172-186

Weibel R. (1995) Three essential building blocks for automated generalization in: Mueller J. C., Lagrange J. P., Weibel R. (Editors), GIS and generalization: methodology and practice, Taylor&Francis, 56-69

Weibel, R., Dutton, G. (1998) Constraints-based Automated Map Generalization, Proceedings of the 8th International Symposium on Spatial Data Handling, Vancouver, 214-224

Weibel, R. (1997) 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, Springer, 99-152

Wertheimer M. (1938) Laws of organization in perceptual forms, First published as Untersuchungen zur Lehre von der Gestalt II, in Psycologische Forschung, 4, 301-350. Translation published in Ellis, W., 1938, A source book of Gestalt psychology, 1923, 71-88

West-Nielsen P., Meyer M. (2007) Automated generalisation in a map production environment - the KMS experience in: Mackaness W. A., Anne Ruas A., Sarjakoski L. T. (Editors), Generalisation of geographic information: Cartographic modelling and applications, Elsevier Science, 301-314

Xuechen L., Bisheng Y. B. (2010) Generating Strokes Of Road Networks Based On Pattern Recognition, 13th Workshop Of The ICA Commission On Generalisation And Multiple Representation, 12-13 September, Zurich

Zhang Q. (2004) Modelling Structure and Patterns in Road Network Generalization, 7th ICA Workshop on Generalisation and Multiple Representation, Leicester, United Kingdom, August 20-21

Zhang Q. (2007) Qingnian Drainage Typification Based on Dendritic Decomposition, in The Cartographic Journal, 44 (4), 321-328

Zhou S., Jones C.B. (2001) Design and implementation of multi-scale databases. In Jensen C.S., Schneider M., Seeger B., Tsotras V. (eds.) Advances in Spatial and Temporal Databases; 7th International Symposium, SSTD 2001, Lecture Notes in Computer Science