Doctoral Thesis

Schematic maps on demand
design, modeling and visualization

Author(s):
Avelar, Silvania

Publication Date:
2002

Permanent Link:
https://doi.org/10.3929/ethz-a-004443857

Rights / License:
In Copyright - Non-Commercial Use Permitted

This page was generated automatically upon download from the ETH Zurich Research Collection. For more information please consult the Terms of use.
Schematic Maps On Demand: 
Design, Modeling and Visualization

Dissertation submitted to the 
Swiss Federal Institute of Technology 
Zurich

for the degree of 
Doctor of Technical Sciences

presented by 
Silvania Avelar
M.Sc. Computer Science, UFMG 
born on October 6th, 1969 
citizen of Brazil

Accepted on the recommendation of 
Prof. Dr. Lorenz Hurni, examiner 
Dr. Marc van Kreveld, co-examiner 
Prof. Dr. Christine Giger, co-examiner

2002
# Contents

Abstract v

Kurzfassung vii

Acknowledgments ix

1 Thesis Overview 1

1.1 Problem statement and motivation .......................... 1
  1.1.1 Design ........................................... 6
  1.1.2 Modeling ........................................... 7
  1.1.3 Visualization ....................................... 7
1.2 Structure of this thesis .................................... 8

2 Schematic Maps 11

2.1 Introduction ............................................ 11
2.2 What is a schematic map? .................................. 14
2.3 History ................................................ 17
2.4 Styles .................................................. 20
  2.4.1 Identifying styles .................................. 20
  2.4.2 Design considerations ............................... 22
  2.4.3 Map variations ..................................... 24
2.5 Map language ........................................... 24
2.6 How are schematic maps produced? ........................ 28
2.7 Examples ............................................... 29
# Contents

3 A Framework for Electronic Schematic Maps

3.1 Modeling a public transport network ........................................ 40
3.2 Model requirements .......................................................... 41
  3.2.1 Queries ................................................................. 41
  3.2.2 Considerations ........................................................ 42
3.3 The transport network data model ......................................... 43
  3.3.1 Overview of transportation features and relationships ............ 45
  3.3.2 Reducing response time and storage requirements to path
       queries ............................................................... 46
3.4 Example of use .............................................................. 48
3.5 Querying the database ...................................................... 49
  3.5.1 Purpose of TNview .................................................. 50
  3.5.2 User interface ........................................................ 51
  3.5.3 Data structure ....................................................... 52
3.6 Conclusions ................................................................. 52

4 Background: Schematic Maps on Demand ............................... 55

4.1 Schematic map as a geometric constraint problem ................... 55
  4.1.1 Constraints .......................................................... 56
  4.1.2 Heuristic search .................................................... 58
  4.1.3 The iterative approach ............................................. 64
4.2 Prior related work .......................................................... 65
  4.2.1 C-oriented line simplification ..................................... 65
  4.2.2 Discrete curve evolution ......................................... 66
  4.2.3 Grid fitting .......................................................... 67
  4.2.4 Top-bottom order .................................................. 68
  4.2.5 Other related work ................................................ 69

5 Generating Schematic Maps ................................................ 71

5.1 General description ....................................................... 72
5.2 Preserving map topology ................................................. 73
Abstract

Schematic special-purpose maps are designed to convey information of limited scope, such as diagrammatic representations of public transport networks. The rationale is that it is more important that users capture the basic structure of the network than to show accurately physical locations on the map.

At present, schematic maps are entirely produced by hand or purely graphic software. This is not only a time consuming process, but requires a skilled map designer. The artist tailors the design to the prospective users and the potential queries they expect to be answered. Currently, there are no cartographic guidelines or orientation to help the design of schematic maps. Automatic generation of schematic maps may improve results and make the process faster and cheaper. More importantly, it would extend the use of such maps to a larger audience, especially to users of transportation systems of many more cities in the world.

This thesis aims to study the generation of schematic maps on demand: a map is automatically generated in response to a selected set of constraints. In the first part of this dissertation, we concentrate on the cartographic design of schematic maps for transportation. We compare schematic maps and classify their characteristics. We also describe aspects to be taken into account when representing schematic routes.

In the second part, we present a framework for electronic schematic maps. The idea is to have electronic schematic maps which can also answer user queries. For it, a data model was developed to describe geographical and topological information of a public transport network. This data model offers a basis for building a transport database with two purposes: to be used in the automatic generation of schematic maps, and to answer location queries of users of the transportation network of the city or region under consideration. As a prototype, we have built a user interface on top of the data model for supporting queries on the transport network.

In the third part, we study the automatic generation of schematic maps from traditional vector-based, cartographic information. By using an optimization technique, the lines of the original route network are modified to meet geometric and
aesthetic constraints in the resulting schematic map. Special emphasis is placed on preserving topological structure of the line network during the transformation process. In general, in order to preserve topological relations of features while transformations are applied, topological information must be computed and consulted during processing. There are no public, generally acknowledged solutions for preserving topology while moving points. We present an algorithm to preserve topological relations using simple geometric operations and tests. We also analyse the convergence of the generation approach and choose a stopping criteria for it. The stopping criteria can give a satisfactory solution, without exploring all possible schematizations of a route network.

The work was applied on a real situation. We generate and design schematic maps using the strategies mentioned above. Our approach provides better results in the creation of an initial map than previous solutions. The generation approach and design information here described are suitable to produce schematic maps on demand.
Kurzfassung


Acknowledgments

I sincerely thank all the people who directly or indirectly collaborated to bring this project to a happy end and also those who taught me valuable skills during the doctorate study.

In particular and first of all, I am very grateful to Lorenz Hurni, my advisor, for the trust and support for the realization of this work. Likewise, I am greatly indebted to my co-examiner Marc van Kreveld for the positive discussions and valuable suggestions for improving this thesis. I also benefited from helpful comments from my co-examiner Christine Giger. In addition, grateful acknowledgments go to the Swiss Commission for Foreign Students for the financial support of my doctorate from 1999 to 2002, and to the Brazilian National Council for Scientific and Technological Development from 1998 to 1999.

Doing research in automated cartography always satisfied me a lot, not only because of the interesting subject, but also because of the nice people I had the pleasure of meeting at events in the area. Special thanks go to Chris Jones and Rupert Brooks, with whom I had helpful email exchanges.

Thanks also to people from the Computer Science Department of the ETHZ, where I initiated this thesis. Jürg Nievergelt pointed me toward this subject matter; Matthias Müller collaborated in the implementation of parts of the visualization module; and Hans Hinterberger and Moira Norrie gave me advice and support.

A big cheers go to all my colleagues and friends of the Institute of Cartography (IKA) and of the Department of Computer Science (INF) for helping me on a regular basis and for making ETHZ such a pleasant place to work. In IKA, I would particularly like to thank Andreas Wipf, Christian Häberling, Christoph Brandenberger, Marianne Rüegsegger, Michael Cooper, René Sieber and Tobias Dahinden (he always managed to get printers and computers working as they should). In INF, Arne Storjohann gave me important feedback on a mathematical question, and Klemens Böhm carefully examined a first draft of this thesis.

Warm thanks still go to all other friends who have also made my doctorate time in Switzerland special. They include Christoph Wirth (he always provided
me with trustable support), Frank Brazile (a busy and wise businessman), Robert Hürlimann (for ongoing support and encouragement), Roger Karrer (a great salsa dancer), and others. It is not possible to mention all here, but I am happy to have each one in my life.

Finally, I thank my family for their support. In special, thanks to my mother, Inêz, for her unconditional support for study and life, and to my sister Solange, for her efficiency and solidarity in many questions.
Chapter 1

Thesis Overview

1.1 Problem statement and motivation

Automated Mapping

Geoscientists and computer scientists have witnessed important developments in acquiring, managing, analyzing, visualizing, and interacting with geospatial data during the last two decades. Results of these technological efforts are today reflected in the proliferation of Geographical Information Systems (GIS) and specific cartographic products, which are employed by a wide range of users in government, commerce and education.

The rise of GIS increased not only the number of those involved in making maps, but also the diversity of automated maps. When making maps, today, cartographic expertise is required, but differently from traditional cartography, since the role of maps has changed and expanded. In the past, maps were designed to be both database and presentation media. Today, GIS and automated cartography have split these tasks, but the link between the two tasks can be better explored. In the new mapping environments, the word visualization requires a different view on map design. Maps are used to present geospatial information, and also to explore geospatial datasets. In on-screen environments, one can store much more data behind the map than on a single paper map, enriching maps even more. In GIS, maps can give the user the graphic environment, the database, the analytical power of GIS, and the benefits of quicker answers for users examination of the data.

The broad applicability of GIS and the increasing market for maps on the Internet have also created a demand for automated maps that adapt to the requirements of the user in the absence of a cartographer. Dynamic, interactive computer-drawn
maps-on-demand can open up the possibility of automatic creation of maps that are customized to suit particular needs. Full automation in map production is though to be an open question. This motivates the pursuit of automation in some stages of the process of map production. Users can get an initial visualization through appropriate computer-based techniques and, if wished, users can later "work out" the result in a creative stage of the design and production process.

Within the existing developments in automated mapping, it is lacking investigation of automatic generation of a genre of map: schematic maps. In this thesis, we will focus on cartographic and computational aspects of the production of schematic maps on demand.

**Schematic Maps**

Schematic drawings of route directions are one of the most common forms of graphic communication. However, this topic has received little attention in traditional mapping and in automated mapping and GIS.

Schematic maps are linear abstractions of functional networks. They are designed to convey only information of limited scope, but which ease their interpretation by concentrating on relevant aspects of information and abstracting from others.

Psychological studies have been performed to test human reactions to schematic maps. Bartram in [Bartram, 1980] shows that, given the choice between written descriptions of networks, planimetrically accurate maps of those networks, and schematic map representations, humans grasp networks and are able to solve problems involving those networks faster and more accurately using the schematic maps. In [Tversky and Lee, 1999], Tversky and Lee show that people are limited in the amount of information and mental operations that they can keep track of, but they are excellent at pattern recognition.

Recently, over 2200 people voluntarily filled out a feedback form describing their impressions of web-based automatic generalized route maps. Less than one percent of the respondents said they would rather use standard computer-generated detailed maps for finding directions to locations within their own metropolitan area. Cognitive psychology has shown that an effective route map must clearly communicate all the turning points of the route, and that precisely depicting the exact length, angle, and shape of each road is much less important [Agrawala and Stolte, 2001].
These psychological results explain why schematic maps are so well accepted for presenting routes in a public transportation network. Besides orientation of transport routes, cartographic schemas can also be used to display the operational status of roads, to show different kinds of events together, such as with gas, water, or electricity mains, and to spatially visualize the nature of a problem [Gordon-Kennedy, 1999]. In this thesis we are especially interested in schematic maps for transportation.

**Schematic Maps for Transportation**

Many cities provide a comprehensive public transport system, often integrating bus routes, suburban commuter train services and underground railways. Information about such transport systems are in general released in schematic maps, which indicate important topological information on transportation, such as connectivity and stops [Monmonier, 1996].

Far from being only a technical subject, schematic maps for transportation affect more aspects of modern life than most would expect. A public transport system is regarded as both a cornerstone of democracy, because it provides equal opportunity for mobility and for participation in the public life, and an investment in the future, because it deals with diversity and choice - people have access to the city as a whole and the choice of using the public transport or not. In complex transportation systems, wayfinding should be supported to a great extent by schematic maps. A schematic map offers a visual tool for communicating spatial concepts for a safer wayfinding task. They can be useful for inhabitants and strangers to the transport system of the city in hand.

To date, easy-to-read schematic transport maps exist only in a few cities in the world. There are several possible reasons for this. One reason that has made this cartographic product so scarce is the lack of documentation and standardization in this subject. Also efforts to elaborate the development of an adequate concept for mapping the transportation system of a city are not done, because, in the amount of subsidy invested to public transport in many cities, funds are not available for the preparation of maps. It is also said that in many large cities, in which buses are the predominant mode of public transport, maps of an entire network may likely to be out of date before they are printed: the more services shown on a map, the more likely some of their routes may change. In smaller towns, the situation may be that local authorities are unable to maintain a professional staff, in general relying upon central state specialists for advice. Finally, another important reason is that currently well disseminated GIS do not offer the possibility for schematization of
networks as an alternative for output data. Schematic maps on demand are also not offered by any current commercial cartographic system.

At present, schematic maps are produced by hand or using purely graphic software. This is not only a time consuming process, but requires a skilled map designer. Computer technology has more to offer in the way of enhancements to the design and production of schematic maps than only a purely drawing graphic software. As generally in cartography, the most basic cartographic problem of the route-based data involves spatial conflict of map elements. With small datasets or simple network systems, spatial conflicts are a simple part of the traditional map design task. When tens or even hundreds of such cases occur, or when the map should be updated and revised often, the problem becomes a crucial issue. Given that we have usually large transport datasets, it is not feasible to do map design by hand. Automatic generation of schematic maps may improve the cartographic process involved and more importantly it would extend the use of such maps to a larger audience.

**Schematic Maps On Demand**

In contrast to the current situation in the production of schematic maps, we aim to contribute to the automatic generation of schematic maps on demand, i.e., a map is generated in response to request from users. Thus, map production changes from supply-driven to demand-driven. Schematic maps could be produced on the fly, each time they were needed, using GIS, and/or the Internet, to display in small screen devices or in specialized interactive video display information sites located, for example, in train or bus stations, tourist bureaus, and car rental agencies.

The idea is to generate electronic schematic maps which can also answer user location queries conveniently using the information stored in a transport database. Users can interact with the schematic map as mapmakers in the map production, and as map readers through location queries. The electronic map must be able to localize in the map the collection of queries that users might expect to be answered. Figure 1.1 illustrates the thesis project.

Users could have the possibility to easily locate an area and information through queries. As example, consider a user that tells a system 'show me a map of where I am'. S/he probably will have a certain purpose or task in mind, such as getting to the next subway stop. S/he may expect to get a quick sketch containing information on public transportation lines and surrounding areas, which the system could infer from the database.
1.1 Problem statement and motivation

The concept of database-cartography can give a number of advantages. We only need to update changes once and they are then updated in all cartographic products using the database. However, in case of schematic maps for transportation, usually, if there is a database of transport services and stops, it is likely to be in a different department or organization from the mapmaker. In many cities, the problem is even to bring together two sets of skills, graphic design and data management, held in different departments, or by different outside consultants [Morrison, 1996].

In this thesis, we will work on three important parts of the presented mapping problem. First, we investigate design characteristics of schematic maps. Then, we focus on solving spatial conflicts in the schematization of lines. We prove our solution theoretically and also practically in a prototype tool for schematizing lines. We use as a basis an existent database containing geographic data of all streets of Zurich. Next, we build a prototype tool to allow viewing the route network and locating user queries in the map. In order to create an efficient and consistent database for this application, a data model was developed to represent a public transport network. The transport database covers two purposes: it contains geographical information of the original routes to be schematized, and it serves as a general information system to users of the transportation network. An experimental database based on our data model was also created.

Specifically, the contributions of this thesis are listed below:

- investigation of design strategies for the production of schematic maps;
• description of a data model with geographical and topological characteristics of a public transport network and of a framework tool to interact with electronic schematic maps;

• development of an algorithm to solve spatial conflicts and preserve topological relationships among features while generating schematic maps;

• implementation of prototype tools to visualize and generate schematic maps on demand and to interact with them.

In the following, we will expand upon each of these results in more detail under the subtitles of the thesis: design, modeling and visualization.

1.1.1 Design

Schematic maps should be designed as intelligibly as possible, in order to facilitate people’s navigation and exploration. Although creating a schematic route map may seem to be a straightforward task, the underlying design of most route maps is quite complex [Agrawala and Stolte, 2001]. Mapmakers perform either consciously or subconsciously a variety of cartographic techniques, including simplification, abstraction and symbolization, to improve clarity of the map and to emphasize the most important information.

A good schematic map requires a lot of work by a graphic artist. The artist tailors the map design to a class of potential users and the collection of queries about directions that users might expect to be answered. There are many aspects of the mapping situation which may need special attention in the design of a schematic map. For example, as will be explained in Chapter 2, in cities over-dependent on one transport mode, more commonly bus networks, the mapmaker will have a more difficult task than in cities with a skillful design of the transport network. Besides characteristics of the transport system, cartographic aspects, such as the amount of simplification of lines and the use of colors, are also important.

We show how to use differing geometric and aesthetic criteria to design a schematic map in Chapter 2. All schematic maps shall have graphic simplicity, while retaining network information and presentation legibility. We also show in the same chapter that the mapping of route-based data poses particular cartographic difficulties related to the effective symbolization of events on routes and to the treatment of overlapping instances of routes.
1.1 Problem statement and motivation

1.1.2 Modeling

A fundamental aspect in a GIS project is the data model, which describes how the geographic reality will be represented in the computer [Halpin, 1995]. We present in Chapter 3 a data model to describe geographical and topological information of a public transport network, in order to generate schematic maps and to answer common location queries from map users.

Some different data models for linear referencing systems have been proposed in the literature, especially due to interests in data sharing and interoperability. Earlier data models, e.g., by Vonderohe et al. in [Vonderohe et al., 1993] and [Vonderohe et al., 1997], were founded upon the single notion of location as a data integrator. But similar scale, same data model and same segment roads were necessary to match spatial data, therefore this was not a satisfactory way to model transport data. Research in GIS for transportation led to a generic data model called dynamic segmentation, e.g., the ArcInfo georelational data model. It integrates the graphics, topology, position, and characteristics of transportation features into a single spatial object [Duecker and Butler, 2000]. However, attempts to use this model in the implementation of data tied to other representations of a linear network is usually problematic [Nielsen et al., 1997, Purtschert, 2001, Avelar and Huber, 2001]. Transportation network models can demand a complex topology which is still not very well covered by most traditional GIS-packages. This will be explained in Chapter 3.

The data model we propose has an object oriented approach which holds transportation features, not their graphical representation, as objects of interest. The data model is more intuitive, and allows the entry of graphics, topology, and characteristics of transportation features separately, facilitating comprehension and maintenance of the database.

We have built a user interface on top of the proposed data model for supporting location queries. The prototype tool enables a user to view the transport network and to explore geographical and topological information of the route network. The aim is to offer a framework for electronic schematic maps to be used in multiple display devices.

1.1.3 Visualization

There exist different approaches to generate schematic maps on demand. For example, Neyer in [Neyer, 1999] and Barkowsky et al. in [Barkowsky et al., 2000] describe a line simplification algorithm with the aim of generating schematic maps.
But line simplification is only one characteristic of a schematic network. Equally important, schematic maps also involve the need of preserving the topological structure of the linear network and other aesthetic design characteristics to be described in Chapter 2.

Elroi in [Elroi, 1991] and Cabello et al. in [Cabello et al., 2001] developed more specific approaches to automatize the schematization process. In the approach of Elroi, a grid is used to fit the line network in the schematic directions, but topological conflicts among line segments can still occur and no explanation is given about how to avoid them. In the approach of Cabello, the topological equivalence between the networks is preserved, but, if certain conditions cannot be met, no schematization is found at all.

In this thesis, the lines of the original route network are iteratively modified to meet geometric and aesthetical constraints in the resulting schematic maps. We use common-sense cartographic constraints in the design of our maps. Users can select the amount of simplification, the schematic directions for lines, and the minimum distance between map features. The generation approach used and other related work are described in Chapter 4. In Chapter 5, we describe the algorithm to solve spatial conflicts and preserve topological relationships among linear features during the network transformation process. The algorithm was implemented and tested in a prototype tool to schematize and visualize a route network.

1.2 Structure of this thesis

The research in this dissertation investigates and contributes to the design, modeling and visualization challenges of schematic route-based mapping. In detail, the thesis is organized as follows.

Chapter 2 describes characteristics of schematic maps and aspects related to their design. Several examples of schematic maps for transportation are also presented.

In Chapter 3, we describe data model requirements for the creation of a public transport network database. First, we present user queries and design considerations of the data model. Then, the transport network model is presented. Finally, we give a tool for interacting with electronic schematic maps and viewing the map answers to user queries.

In Chapter 4, we give the necessary background to understand how iterative techniques can be used to generate schematic maps, and present a brief review on the domain of schematic maps on demand.
Chapter 5 presents the generation of schematic maps on demand. We show that the iterative approach used leads to topological errors. We give an efficient algorithm based on simple geometric operations and tests to preserve topological relations among linear map features during the schematization process. Proof is also provided. We still present the convergence evaluation of the iterative algorithm and investigate a stopping criteria for it.

In Chapter 6, the ideas in this thesis are demonstrated in two real-world examples.

Finally, conclusions, open problems, and future research directions are given in Chapter 7.
Chapter 2

Schematic Maps

Transportation has been a subject of maps for centuries. However, the cartographic literature offers little guidance to a map designer seeking tactics or practical ideas for representing elaborate route data schematically. The need for design strategies based on sound principles grows as GIS practitioners embrace the measured route as a base on which features can be mapped. This chapter aims to contribute to the design challenges of schematic route-based mapping. We offer information about schematic maps, as well as insights into how to create such maps for different contexts. We describe cartographic rules for symbolizing route-based data and give examples of schematic maps.

2.1 Introduction

Cartography involves the transformation of data into representational visual models that communicate, enlighten, integrate and persuade. The communicative purpose of a map determines what information should be kept and what information can be eliminated. Mostly, cartographers attempt to create precise miniature replicas of selected land features, but inevitably some distortion occurs. For instance, road widths and river sizes are not drawn to scale on most maps as the roads and rivers would not be visible if they were. This has traditionally been seen as a relatively unimportant side effect of mapping [Dorling, 1996]. Every map has, to some extent, been simplified or generalized [Campbell, 1998]. In some special cases, the geometric accuracy in the map will be less important than functional relationships between mapped features. To serve this purpose, mapmakers sometimes distort the map’s geography to show relations that are more meaningful to the map users [Muehrcke and Muehrcke, 1997].
Cartograms

Maps are called cartograms (or 'Anamorphosen', in German) when distortions of area, and occasionally of shape or distance, are made explicit and are seen as desirable [Dorling, 1996]. Typically, places on a cartogram are drawn so that their distortion is in proportion to some value attached to the map data. For example, conventional maps can be seen as land area cartograms, because places on them are drawn in proportion to their land areas. The distortions reveal information about the places that would otherwise be difficult to observe [Campbell, 1998].

Cartograms are produced for a variety of purposes. In atlases, cartograms are often used for their ability to provoke. For example, a cartogram where areas are drawn in proportion to the wealth of people living in the represented places shows a dramatic picture. In human geography, area cartograms commonly have size of places in proportion to their human population, such that a more socially just form of mapping is produced by giving people equitable representation in an image of the world. Considering distances, in many situations we may care more about the time or the cost of travel to get to some place than about the distance in usual terms. To serve this desire, real world distances can be distorted in linear or distance cartograms, in order to reflect the required measure to get from one place to another. In route maps, routes can be distorted in distance and shape proportionally or not to some value to help people find their way.

Route Maps

Throughout this thesis, we are especially interested in route maps. Route maps are one of the most common forms of graphic communication. One can hardly think of a more basic kind of map than a drawing of route directions. Often route maps have their content sharply limited, being diagrammatic, rather than realistic. This is possible because a route is a path on which awareness of cardinal, three-dimensional space is unnecessary for successful navigation so long as one follows the correct route in the correct direction [Gordon-Kennedy, 1999]. This situation leads map designers to break cartographic rules and tailor their route maps to specific requirements for convenient orientation of map users.

Among the more effective route maps are highly generalized diagrams\textsuperscript{1} por-

\textsuperscript{1}The terms diagrams, diagrammatic maps, and map-like diagrams are used interchangeably in this thesis. Although, when these terms are used, they give the idea of a generalized content, while map means the general category. A map, according to the definition of ICA (International Cartographic Association) from 1995, is a symbolized image of a geographical reality, representing selected features or characteristics, resulting from the creative effort of its author's execution of
traying underground and transit rapid systems. There is a special tolerance for, and acceptance of, geographical inaccuracy to represent underground routes, but the structure of the route network has to be essentially the same as the structure of the real route directions [Tversky and Lee, 1999]. The presentation of underground route maps as a cartoon illustrates how the idea of stretching space is intrinsically acceptable to a large audience. Such diagrammatic maps, by virtue of their sparseness and imperfection, provide a perceptual stimulus that supports and facilitates the formulation of mental objects without overpowering the user imagination with the completeness of the concrete visual world or with details shown in non-diagrammatic maps.

Underground maps depict routes as a compromised spatial and logical transformation. The only relevant data for users, once the structure of the network has been entered, are the general directions to be taken and the proper points of transfer. Distortion of spatial relations and exclusion of details are used to simplify the complexity of the route system for presentation. As a consequence, such maps often convey qualitative spatial concepts adapted to common characteristics of mental knowledge representation [Barkowsky et al., 2000]. Some authors, like [Dorling, 1996] and [Monmonier, 1996], classify underground route maps as linear cartograms, but this can be questionable, since distortions in the map are not in proportion to a measurable feature. Underground route maps are schematic maps.

There is a significant difference between making accurate conventional maps and making diagrammatic maps. In general, a conventional map does come with a set of rules, which, if broken, reduces its effectiveness. In diagrammatic maps, such as schematic underground route maps, some cartographic conventions and shapes already known should be respected, but the emphasized information can be presented as a cartoon geography. Diagrams are pictures – pictures of information. When making diagrammatic maps, meaning should be brought to the right amount of information, which should be presented into an aesthetically pleasing whole. The problem diagram designers face is to keep the picture from overwhelming the information it represents [Holmes, 1993]. Graphical excellence consists of complex ideas being communicated with clarity, precision, and efficiency.

The remainder of this study about schematic maps is organized as follows. We describe characteristics of schematic maps in Section 2.2. A brief outline of the historic development of such maps is given in Section 2.3. We suggest map design tactics and present styles of schematic maps for transportation in Section 2.4. Elementary cartographic properties of route mapping can be very useful for this study, therefore, in Section 2.5 we describe general symbolization and rules for choices, and is designed for use when spatial relationships are of primary relevance.
route-based mapping. At the end of the chapter, we give examples of design of schematic maps used to illustrate different transport data.

2.2 What is a schematic map?

Moving from the physical reality to a pictorial representation of it, we can obtain a mild abstraction of the reality by taking a visual image of it, e.g., a photograph. Moving a few steps further towards concept formation, we may get a topographic map in which objects from the real environment have been identified, interpreted and symbolized, and spatial relations are maintained. Further abstraction may lead to sketches and schematic maps. See Figure 2.1.

People without particular skills or credentials make sketches for their own use or for others with their own interpretation and reconnaissance of the landscape to communicate specific geographical ideas. Cognitive psychologists have shown that people use some kind of mental map when they deal with space in tasks such as scene description, navigation, and spatial reasoning [Timpf et al., 1992]. Sketches closely mimic the way we store information about our physical environment, as mental maps. Few people’s mental maps will correspond precisely with cartographic maps, but it is the mapped world, not the map, which they are trying to understand.

There is no sharp boundary between sketches and schematic maps [Freksa et al., 2000]. Sketches usually have close correspondences to verbal de-
2.2 What is a schematic map?

Schematics about spatial features. Verbal descriptions are interpreted and stored as mental maps. Schematic maps are related to the transformation of mental maps of verbal directions to a map [MacEahren and Johnson, 1987]. They can also be obtained by relaxing spatial and other constraints from more detailed maps [Casakin et al., 2000]. Schematic maps differ from sketches in that they are meant to represent a certain part of the environment completely at a given granularity level. Sketches are more typically about a small set of features or about a single route. However, schematic maps may be incomplete and sketches may be unusually elaborate [Freksa et al., 2000].

A schematic map is an easy-to-follow diagrammatic representation based on highly generalized lines which is in general used for showing routes of transportation systems, such as subways, trams and buses, or for any scenario in which streams of objects at nodes in a network play a role. Examples of such other scenarios are cartographic schemata for gas, water or electricity mains, and tourist city maps.

Psychological studies have been performed to test human reactions to schematic maps. They show that, given the choice between written descriptions of networks, planimetrically accurate maps of those networks, and schematic map representations, humans grasp networks and are able to solve problems involving those networks faster and more accurately using the schematic maps [Bartram, 1980, Agrawala and Stolte, 2001]. It has also been shown that people are limited in the amount of information and mental operations that they can keep track of, but they are excellent at pattern recognition [Tversky and Lee, 1999].

In this thesis, we will focus on schematic maps for showing routes of public transportation systems. However, most of their characteristics can be extended for other areas of application. In this type of map representation most features not directly relevant for using buses, underground trains, etc. are omitted [Barkowsky et al., 2000]. The map concentrate on stations, the lines connecting them, and some typical features helpful for the overall orientation within the transportation system, being that it is often used in conjunction with timetable information.

If you try to superimpose a schematic map on a standard map, you will find that the positions of stations are not at all correct. What is correct is the representation of the network. The map tells the user what transport line to take from point A to point B, and where to change lines if necessary. The crucial point is that in this one respect the schematic map is completely accurate. It succeeds in capturing an important pattern in the geography of the transport network system, which is called a topological pattern. Therefore, we say that such a map is topologically equivalent.
with a classical map that obeys the reality more closely.

In two dimensions, topology is the mathematical discipline which studies properties of figures that are unchanged by stretching or twisting the surface on which the figures are drawn [Devlin, 1994]. Topology is, therefore, sometimes referred to as 'rubber-sheet geometry'.

In geometric terms, the schematic map is hopelessly inaccurate: the length of the lines has no correspondence to the actual length of the trajectory, and the physical track is not as straight as it is drawn on the map. Function dictates form, and a more accurate map in the usual sense would not work as well. For recognition purposes, direction and distance can be only roughly preserved, while topological information of the line network has to be preserved.

Schematic maps have all routes usually drawn as straight lines with cartographic microcrenulations removed. The lines vary in direction only via fixed, stylized angles, commonly of 45 and 90 degrees, or of 30, 60 and 90 degrees, though in some schematic maps lines are merely simplified with arbitrary, but few directions [Morrison, 1996]. Overlapping lines are in general separated by a minimum legibility distance, which can either be zero or a constant chosen for the map. Usually, but not necessarily, adjacent schematized lines have smooth, artistic, circular arcs around bends, preserving their graphic proximity distance for the greatest length possible (see example in Figure 2.7 at end of this chapter, top left corner, where the lines curve in between two white cylinders). We can also observe that in some complex transportation systems straight network lines are often not too long and a small number of breaks or changes in line direction can be added to provide a better visualization of the lines and add a sense of the original geometry.

Another important characteristic of schematic maps is that they do not have a constant scale factor for the entire map. In public transportation maps, the scale is relatively large for the inner city, where many routes converge and connect; stops in the central business district might be only four or five blocks apart, and a larger scale is needed to accommodate more route lines and station names. In contrast, toward the fringes of a city, where stations are perhaps more apart, scale can be smaller because mapped features are less dense [Monmonier, 1996].

In some schematic maps, all lines are presented in one color and the route of a particular network is traced by number only. In other maps, contrasting colors differentiate the various routes. The range of coloring methods can be as large as that for conventional non-schematic maps. We give more details about such design styles in Section 2.4.

In reality, differing geometric and aesthetic criteria can be used to design a schematic map. It is common to find differences in style, but all maps share the
need for graphic simplicity, while retaining network information content and presentation legibility.

2.3 History

Underground Beck Diagram

The most used example of schematic maps are underground maps. This is not without reason, because the London underground map [TfL, 2001] is the pioneering diagram, designed in 1931 by Henry C. Beck, a 29-year-old engineering draughtsman.

It took two years of persistent efforts by Beck before his now-familiar map to be accepted for publication. Even then, the Underground Publicity Department produced the map only in small numbers. Their fear was that the total abandonment of geographical accuracy of the map would render it incomprehensible to the majority of underground travelers. But they were wrong. The public loved it. People couldn’t resist the helpful character of the schematized routes, appreciating indistinctively that its designer was concerned for their information needs and not for novelty for its own sake. Thus, by the end of its first year in use, a larger version was posted all over the system [Garland, 1994].

Without the need for any explanation or training, the general public not only coped easily with their first explicit encounter with a genuinely topological representation of the underground network, but they recognized at once its advantages over the more familiar geometric depictions. Once the basic linkage had been absorbed and its litany learned, for example, that Leicester Square was ‘below’ Tottenham Court Road, Oxford Circus to its ‘left’, Goodge Street ‘above’ it and Holborn to its ‘right’, places were filled into the newcomer’s mental map sooner or later.

In fact, Beck conceived the diagram to help underground travelers to get to the right station, make the right connections and get off at the right destination, but the map quickly became more than that. The diagram offered a unique visualization of what was probably the most intricate pattern of rail connections in the world. Before the diagram, it was hard to make sense of the intricate web of connections of the complex system of London (see Figure 2.2), which is not a grid city like New York, or a radial city like Paris. Garland in [Garland, 1994] states that above any consideration of the diagram as a navigation aid is the optimistic vision it offered of a city that was not chaotic, in spite of appearances to the contrary. Gordon-Kennedy [Gordon-Kennedy, 1999] also points out the promotional purpose of this type of map-like representation in making routes appear efficient and direct.
Figure 2.2: 1932 underground map of London (above), and the first card folder edition of Beck's Diagram (below) issued in January 1933. Source: [Garland, 1994].
In Beck’s genesis of the diagram pointless or distracting variations were not incorporated, so he did not loose control of the essence of the diagram. The only surface feature included in the map and one of great importance was the stylized representation of the River Thames. The diagram reflected this in its unemphatic display of the central area, where no single feature was dominant. Equally important, in order to achieve a clear, comprehensive array of features, the central area was enlarged in relation to the outlying regions. Thus, purposeful and skillful distortion was of essence.

The diagram has changed very little since 1959, the year of Beck’s last design (notice that twenty-seven years were dedicated to improve the graphic design). Although some tentative changes on it were done, for example, for representing interconnection stations with outsized diamonds and for using thickened route lines, such changes had the effect of altering the diagram markedly for worse.

The longevity of Beck’s diagram is a testament to its utility and aesthetic appeal, but of course the diagram is not perfect. It may be argued that it was sacrificed just too much geographical resemblance in the cause of clarity; and it has also been accused, by some, of presenting an oversimplified view, not only of the network, but of London itself. However, neither of these criticisms can diminish its shining example nor take anything from the personal achievement of its inventor, who gave a new approach to mapping public transportation and an important contribution to the development of graphic design in the twentieth century [Garland, 1994].

Meanwhile, making a simple, schematic map which focuses on the route network, not on the environment as a whole or on absolute directional relationships, is not a new idea. The concept of a cartographic line as a route is very old in cartography.

Earlier Maps

Most of the earlier route maps were of the strip map subclass, which depicted routes in linear forms [MacEahrenheit and Johnson, 1987]. One example of such strip maps are medieval maps for coachmen on which the roads were drawn as a straight line, just like the road stretching out in front of the coach and horses [Holmes, 1993]. It was the compass arrow that moved to show changes of direction, while the road kept relentlessly on up the narrow pages of a concertina-folded pack that was unfolded as the journey proceeded. On either side of the road were shown landmarks and intersections that would be passed along the way, but nothing out of the coachman’s view was included. The map, therefore, served for that particular journey alone, and could be used in both directions. Another often cited cartographic example of antique route map are the Peutinger Tables or Itineraries, which are itinerary
route cartograms made by Romans in the 1st century [Goss, 1993]. The map was originally a long, narrow parchment roll, showing schematized imperial roads in strip format covering roughly from Southeast England to present day Sri-Lanka [Siebold, 1998].

Figure 2.3 shows a strip map of January 1959, from Union Pacific Road, USA. Strip format maps decreased in popularity, because they were for single purpose trips only, and were ill suited to performing route planning, especially for intra-urban public transit [MacEahren and Johnson, 1987].

![Figure 2.3: Example of a strip map, which represents a route in its elementary form. Source: [Gordon-Kennedy, 1999].](image)

**2.4 Styles**

Creating a schematic route map may be seem to be a straightforward task, although, the underlying design of such maps can be quite complex. Mapmakers use, consciously or subconsciously, a variety of cartographic generalization techniques, including simplification, distortion, and displacement, to improve the clarity of the map and to emphasize the most important information. This type of generalization is prevalent both in quickly sketched maps and in professionally designed schematic route maps, such as appearing in subway schedules, and print advertisements [Agrawala and Stolte, 2001].

**2.4.1 Identifying styles**

The principles of the London underground map have been applied in the design of schematic maps for showing the transportation systems of many other cities and countries. According to Petchenik [Petchenik, 1974], style emerges when many examples have some recognizable and widely accepted visual similarity. Morrison in
2.4 Styles

[Morrison, 1996] compared schematic maps of different western European cities, in order to identify styles. He suggests a map classification which we briefly describe:

**Classic style**: This is the style in which one line is used to represent all the services of each transport mode on a street, and the routes of individual services are indicated only by writing the service numbers alongside these lines. Morrison wanted to call it as British style, but, in virtue of the current existence of other different map styles in Britain, the 'classic' designation was preferred. It is mostly found in British, Italian and Portuguese towns.

**French style**: Identifies every service of each transport mode by a different color. The service numbers appear in general at the two termini. This is the only style used in France, and it is also used in most cities of Switzerland, Belgium, and in some towns of Italy, like Venice.

**Scandinavian style**: It is the same as the classic style, but applied to several subdivisions of the transport network separately. The sub-divisions are chosen such that usually only two or three different lines appear on one street. Each line uses a different color. The style is used in larger cities of Scandinavia, in Germany, Austria, and some cities of Spain.

**Dutch style**: Similar to the classic style, but with a different symbology for each transport mode. Often trams are represented by a double line, just like tram tracks in the street, and railways are represented by the traditional cartographic symbol of a double line with a broken filling. This style is mostly used in the Netherlands.

Other styles may still be found or invented in the broad scope of possible cartographic schematic representations of a dataset. One could use color to distinguish the transport modes rather than the transport services. Or colors could indicate the general direction or destination of the routes, such as red for services passing through the city center, yellow for 'transverse routes', green for 'peripheral and other routes', blue for 'night routes', black for representing underground and suburban train lines, etc. Important is the legibility of the map, which should be always tested. Some maps succeed in showing many services of one transport mode legibly by making some compromises with the principle of one color per service, in that, for example, all privately owned services are presented in a same color. In case there are too many services and we want to keep a limited number of well distinguishable colors, we can introduce variation in line character in addition to color. Obviously, schematic maps can be more legible if the transport network itself is skillfully designed.
2.4.2 Design considerations

We now identify several aspects that should be taken into consideration when designing a schematic map. Considering the previous style classification, we have to observe mainly aspects related to characteristics of the transportation system to guide the choice of map style. These aspects include the number of transport modes, such as bus, tram, train, etc., number of services of each transport mode, predominance or not of one transport mode, existence or not of overlap between routes, and if there are variances in routes [Morrison, 1996].

Cartographic aspects of the representation of transportation routes may also need attention in the design of schematic maps. These aspects include the amount of base map detail which is required, the need to emphasize names of terminal, the consideration of appropriate background colors, the need to insert panels containing the services passing at interchange stations, and the need to use insets.

There are still other aspects, according to the mapping situation, which can be relevant for the choice of map style. Some examples are: the users can have certain preferences which may determine the map style, the route pattern of the city can be more adequately presented in a certain style, the amount of resources available to produce the map, and if there is any restriction on the use of colors, mainly for maps to be posted outdoors. Considering this last aspect, the fading effect of the sun, especially in sunny countries, may require special inks, resistant to fading in the sun.

An infinite range of possible colors is available, but this does not mean the colors are all distinguishable. In cases in which routes appearing in similar colors meet, confusion could result. The system of service numbering could help to solve the problem of many routes meeting, like in city centers. For example, it would be natural to call variants of a basic route number 1 as 11, 12, 13, 14; or 1, 1A, 1B, 1C; or 1B, 1G, 1H, 1T, using the initial letters of the destination points.

Considering transport modes, it has been observed that schematic maps can be more appropriate for representing some transport modes than others [Morrison, 1996]. In the particular case of buses, transportation institutions of several towns reported that bus travelers complained they could not understand the schematic map and hence could not use it. The reason seems to be that the lines on the schematic bus map cannot be easily related to the reality of the street plan of the city which forms the bus traveler’s mental map. S/he perceives bus routes as following the actual streets, and is very aware, while traveling, of the sharp turns in the bus route, which may not correspond to the straight lines and circular arcs on the schematic map. Thus, schematic maps can be preferable when they correspond
to the traveler's perception of the routes as straight lines. In general, if the traveler is able to obtain some clues to his/her true location and direction of travel by looking out of the window, e.g., the sun, major rivers, and motorways, s/he may be disturbed if the schematic map shows things differently. Schematic maps are, therefore, usually more acceptable in this order: for underground railways, surface railways, trams and light rails, and buses.

Considering buses, the awareness of the transport user will be relatively less important in cities where bus stops have names clearly marked upon the routes, so that one can follow the travel progress by reading the names of stops. In these cases, successful schematic bus maps are found, for example, the bus map of the city of Porto shown in Figure 2.8. We call attention to the fact that, in general, these successful schematic bus maps include a simplified version of the city plan on the background, which also helps the travelers to orient themselves. Such schematic maps contain simplified streets and landmark features of the traditional map of the city in hand on the background, and show the schematized bus routes usually in classic style. Though this map design is more commonly used for buses, we can also find schematic maps with similar design presenting other transportation modes, for example, the subway map of Madrid illustrated in Figure 2.9.

Still in case of buses, we add that, in informal communication, some people affirmed that, having the choice, they avoid to use buses, because they find them confusing. Being in an area that they are not familiar with, and if the bus stops are not clearly labeled in big lettering (or announced effectively on the bus) – as it has been the case in the overwhelming majority of bus trips in most cities of the world –, then it can be really difficult to work out where to get off. Especially at night, when it is hard to see the street names, even if there are any. One has to keep on asking the driver – "Is this X Street yet?" – and s/he says, "I'll let you know when we get there", but as time rolls on the bus traveler begins to wonder whether the driver has simply forgotten. Traveling by bus in these conditions may be hard, unless there is a schematic map showing the bus routes and the territory. For these reasons, schematic bus maps containing no background at all or only a few supportive information in the background, like water features, can make things difficult when navigating. The amount of background information has to be carefully chosen. In general, schematic maps do not contain much information about the terrain in the background, nevertheless hill shading or spot heights of major features could also be included.
2.4.3 Map variations

The styles to design a schematic map for showing collective transport in a city can be considered for various types of schematic maps. Besides the usual schematic maps, which contain generalized lines to represent the transport network routes, there are also variations in the schematic maps related to the extension of the area shown and to the aim of the map. These variations include “octopus maps”, which aim to show public transport traffic extending from a single station [Morrison, 1997]; and “thermometer diagrams” or “string of pearls”, which are composed of a straight or stylized line diagram showing one individual route, e.g., for showing a single line inside a transport service vehicle, or for when the line path is not considered relevant to be communicated, but only the order of stations. Another variation of schematic map is airline or ship route maps, which are printed on a background representation of the terrain. These maps eliminate completely details from network routes, and the routes are displayed as arcs rather than as straight lines [Elroi, 1991]. The paths of the lines seldom coincide with the actual routes between the cities they serve, because real routes vary due to weather conditions and other factors [Campbell, 1998].

Recently, in London, conventional local area maps containing streets and bus routes on and near each bus station were changed by so called “spider maps”. These maps are based on spider diagrams, which represent main ideas in the middle and details on branching lines. The new spider bus maps are tailored to show an inset centrally located, containing a geographically accurate map of the area around a bus stop, and an outer schematic representation of the bus network destinations and connections possible from the local station. The central inset contains streets, landmarks, and other bus and underground stations nearby the mapped station, so that bus travelers can have enough information to know where they are and where they can get to from that point. The outer schematic lines have the same classic style of the London underground map [TfL, 2001]. Figure 2.15 shows an example of spider map.

We describe following basic concepts of cartographic symbolization, which are important for using the described map design styles or for creating new ones.

2.5 Map language

We presented different styles and types of schematic maps in the previous section. The purpose of this section is to provide an overview of cartographic symbols for the communication and visualization of route data. With that, we want to provide
some awareness on design issues which concern the ways in which map symbols can be manipulated to affect the impact of a map.

Once geographic features and data to be represented in a map have been, in general terms, identified, interpreted, simplified, and classified, it is necessary to choose an appropriate graphic representation or symbology for the information. The information must be presented in a well organized manner, in order to allow an optimal perception of the map contents [Hurni and Leuzinger, 1995]. Before assigning a map symbology, it is therefore important to have a good understanding of the information to be mapped. Symbols have characteristics that can be manipulated to suit the category of data being mapped.

Graphic symbols can be classified according to the type of spatial objects they represent. In two dimensions, this leads to the familiar division between point, line and area symbols. Clearly, the type of symbol chosen depends upon the degree of generalization of the phenomenon being represented. The 2D graphic symbols can be modulated graphically in various ways to help in communicating different types of information [Jones, 1997]. The primary graphical characteristics of symbols which can be varied are called visual or graphic variables. Following Bertin [Bertin, 1983], the visual variables of symbols are size, shape, orientation, pattern texture, color (hue), and color value (brightness and lightness).

The mapping of data based on a linear frame of reference poses particular cartographic difficulties that have not been developed thoroughly in the mainstream cartographic literature [Gordon-Kennedy, 1999]. Besides using generalization techniques, mapmakers handling route-based data will face some or all of the following requirements: display coincident point and line events, display multiple event attributes, labeling of relevant features, and time (some of the data associated with routes is most valuable when analysed temporally).

For Gordon-Kennedy in [Gordon-Kennedy, 1999] a specification for cartographic representation of route-based data begins to emerge when one considers three central characteristics of the route-based mapping problem: 1. the spatial constraints of lines, i.e., the events occurring on routes should be constrained to preserve topological accuracy and to maintain good graphic association for the map reader; 2. the properties of events, e.g., the event data have characteristics inherent from the route data, such as directionality, sidedness, punctual or linear representation, and chronology; and 3. the properties of the thematic data for cartographic treatment of lines.

As thematic data, the attributes of events encompass all classes of measurement: nominal, ordinal, interval and ratio [Robinson et al., 1995]. Continuous events, such as pavement condition, can have attributes like type of material, thickness,
or number of lanes, any of which could be classified to represent the pattern of values. Variations in thickness, texture, and color are used to indicate thematic data on lines.

<table>
<thead>
<tr>
<th></th>
<th>Nominal</th>
<th>Ordinal</th>
<th>Interval/Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td></td>
<td></td>
<td>1.0 2.0 3.0 4.0</td>
</tr>
<tr>
<td>Texture</td>
<td></td>
<td></td>
<td>None recommended</td>
</tr>
<tr>
<td>Color</td>
<td></td>
<td></td>
<td>None recommended</td>
</tr>
</tbody>
</table>

Figure 2.4: Linear symbols for mapping route-based data [Gordon-Kennedy, 1999].

Tables of graphic techniques by data type are proposed by Gordon-Kennedy [Gordon-Kennedy, 1999] as an initial framework for classifying the application of graphic variables to the route data. Figure 2.4 shows how linear symbols can vary by size, texture, and color to represent most types of attribute data. Line size, texture, and color are widely understood as graphic variables for showing nominal and ordinal data, such as road classification. Line texture and color are not, however, traditionally considered suitable for interval and ratio data, where line thickness has been the convention.

The cartographic literature offers some general guidance for representing route linear data, but is apparently barren of any guiding cartographic principles for the depiction of point data on routes. Figure 2.5 shows how point symbols can vary by size, texture, and color to represent most types of attribute data [Gordon-Kennedy, 1999]. Interval and ratio data also present some special difficulties. It is difficult to envision how a point symbol’s texture might change on a continuous scale to represent interval or ratio data values. Color could be varied in hue or value on a continuous scale, but the effectiveness of such point symbols is also questionable.

Considering the representation of coincident and overlapping routes, generally mapmakers can deal satisfactorily with overlap of up to three or four routes. It can be feasible to show all routes individually, without using any special symbolization, but for each case there is a maximum number of routes to be shown legibly side-by-side. The clarity of the map can also be improved by deliberate distortion, like
2.5 Map language

<table>
<thead>
<tr>
<th></th>
<th>Nominal</th>
<th>Ordinal</th>
<th>Interval/Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size</strong></td>
<td>A B C</td>
<td>1 2 3</td>
<td>1.0 1.5 3.0</td>
</tr>
<tr>
<td><strong>Texture</strong></td>
<td>A B C</td>
<td>1 2 3</td>
<td>None recommended</td>
</tr>
<tr>
<td><strong>Color</strong></td>
<td>A B C</td>
<td>1 2 3</td>
<td>None recommended</td>
</tr>
</tbody>
</table>

Figure 2.5: Point symbols for mapping route-based data [Gordon-Kennedy, 1999].

Central areas can be enlarged relative to suburban areas.

Figure 2.6 depicts examples of how coincident and overlapping events on point and line datasets might be displayed. Displacement, the setting off of additional symbols parallel to the route, can effectively show the clustering of coincident events, but this tactic would probably work poorly with more than a few cases. Rather than displace symbols to the side, one could simply let them superimpose where they coincide. Carefully planned symbology might allow readers to recognize the component elements, but the question is how far this can be taken. A third approach to coincident events might be to use a symbology that works in admixture: symbols that when superimposed create a new symbol that the map reader can de-compose. This might serve well for special designed textures, but color is another matter. Can we presume map users would read a purple symbol as the coincidence of blue and red symbol? The concept of linear clusters, aggregations of either discrete points or short segments, may also need development. A fourth option is to use a distinct new symbol for instances of coincidence and overlap, a solution of limited usefulness.

There is still a need for principles of design that address the route-based symbology, especially for interval and ratio data types, in order it is effective the treatment of symbol conflict, coincidence, and multivariate symbols.

Psychophysical cognitive studies have been a central theme of contemporary academic cartography, but surprisingly little research can be found on route symbology. Some questions about the visual perception of route data are: Is it not plausible to expect readers of route-based maps to visually estimate the lengths of segments? Will a bright yellow element be judged as accurately as a black one? What is the effect of line thickness?
Symbology for Displaying Multiple Events

Point Symbols

| Displacement | A   | B   | A+B |
|             | ●   | □   | ■   |
| Superimposition | A   | B   | A+B |
| Admixture     | A   | B   | A+B |
| New Symbol    | A   | B   | A+B |

| Line Symbols |
| A | B | A |
| A | B | A |
| A | B | A+B | A |

Figure 2.6: Some of the symbology options for representing coincident and overlapping instances of route-based data.

Both precise data representation and visual clarity are top priorities to the choice of an adequate symbology to represent the route data. We recall that for each mapping problem there will be a unique combination of elements at work.

2.6 How are schematic maps produced?

Schematic maps for public transportation have persisted since the London underground map, but, as we mentioned previously, only a small amount of work has been written about them, and the process by which they are produced has not yet been completely codified in cartography. In the following, we sketch out the methods possible to be used to schematize a map.

We can categorize the schematization methods into three main classes: manual, assisted, and automatic. In the first method, the mapmaker produces sketches by hand to search for the most pleasing graphical solution, adjusting and readjusting the network until the map has reached a satisfactory state without loss of topological information of the network. This is surely quite labor-intensive and unpractical. The next method is the one more currently used. It applies a purely
draughting software to assist the map drawings by computer. In general, the original road network is scanned or digitized, then used as background to the drawing and design of the new schematized lines. This is still a procedure of trial and error attempts, but results can be obtained quicker than in the manual method, attempts can be stored, and output to paper can be easily arranged. The method, however, requires just as much visual scrutiny and iteration as the manual method. The third method involves the use of specific implementations for the automatic generation of schematic maps. The schematization process should be broken into a set of tasks that can be used to implement the schematization, and thus make it more easily available. This method has also the advantage of making schematic maps more effective, because of the graphic and analytic possibilities of a vector-based system.

Elroi in [Elroi, 1988b] introduces an untested method to make schematic maps, which he calls mechanical method. He proposes to use a simple device, whereby the planimetrically correct network is duplicated with colored elastic strings over a matrix pegboard. Pegs are then inserted into each resultant polygon and shifted around for the best result. This method is obviously not practical. Elroi agrees that it may be difficult to implement such a device, but he emphasizes that it provides a regular matrix background, as well as the ability to eliminate line details and to help to assure that topological characteristics of the network are maintained.

2.7 Examples

We show here several examples of schematic route maps. They reflect only a tiny part of the effort map designers have put into creating an alternative basis for schematizing maps.

There are traffic schemes used in a rather general sense, such as rapid transit, bus, and airline diagrams, and also those under specific categories, like detours and departure notices at displays of the operational status of a transportation system. Figures 2.7, 2.8, 2.9, 2.10, 2.11 and 2.12 illustrate different schematic route maps showing the public transport network of the cities of Zurich, Porto, Madrid, Copenhagen, Amsterdam and Baghdad, respectively.

The tram map of Zurich in Figure 2.7 presents all tram lines in the French style and the bus lines in the classic style. The Porto bus map in Figure 2.8 has less aesthetic treatment and relatively less geometric simplification than the public transportation map of Zurich. The routes in the bus network of Porto are designed also in the classic style.
The Madrid subway map in Figure 2.9 presents transport lines simplified only, i.e., they do not have fixed schematic directions. The city plan included on the background contains streets, traces of vegetation and shops (probably the commercial sponsor of the map). Unusually, the schematic map contains also a graticule, a graphic scale, and a north symbol.

In Figure 2.10, we see another example where bus routes are merely simplified. The Copenhagen map of bus routes is designed in the Scandinavian style. The map brings more information to bus users than the Madrid map. All streets containing routes have their names identified, as well as main streets. Regions of the city are identified by names and numbers, e.g., 'City' is '1'. The map includes also vegetation, reference places and some insets containing the list of routes passing at certain stations.

The map of Amsterdam in Figure 2.11 has the Dutch style. An interesting design feature of this map is that many services are represented by using only five colors. Colors indicate the general direction of routes. The red lines show the routes to various tourist sights of Amsterdam, they go from the center of the city to the west. The blue lines pass through the city center to east. Yellow lines go from east to west via 'Ceintuurbaan', and green lines go from east to west via 'Weteringschans cq Dam'. Notice that the visual variable 'size' is used to indicate coincident paths in routes of the same category.

In the map of Baghdad in Figure 2.12, there is no identification of the transport lines on the map, only station names. The interconnection of lines is not easily understood. We suppose that regular users may have had preferences taken into consideration in the design of the map, because a visitor could find it confusing, unless s/he has time to read the explanation, which appears in Arabian and English in the right down side of the map.

Stylish failures are also a common outcome of the graphic design of route diagrams. The example in Figure 2.13 reveals the difference a good design makes. The purpose of the map is to show the location of a hotel in front of the 'Oerlikon' train station in Zurich and how to reach it. But the hotel name and address are not identified in the map. Also not all street names are identified, being that neither all streets presented are main streets nor they have paths schematized in their natural directions. Space is poorly allocated, much of the paper is given to create an elaborate but false appearance of systematic order. A user in the 'Hohlstr.' could be confused how to reach 'Wehntalerstr.' or the street before, 'Schwamendingenstr.'. Garland in [Garland, 1994] considers as an effective force for map quality control the wise practice of old maps of putting the names of the people responsible for the cartographic design on the map, what can also be a sign of pride for them.
A stylized thermometer map (or string of pearls) is shown in Figure 2.14. The map presents a view on two Japanese subway lines and their connections. The loop of the Yamanote Line contains 29 stations and it is connected by all of the 10 subways and private railway lines. An interesting legend was created to show connections. The main interchange stations on the counter-clockwise circuit after Tokyo Station are: Akihabara, Ueno, Ikebukuro, Shinjuku, Shibuya and Yurakucho.

Figure 2.15 shows an example of spider map created for the bus network of London. Spider maps are considered clearer, but many people reported they preferred the old local area maps, because they regularly used the bus shelter maps to locate the best pedestrian route somewhere or find a street or landmark not immediately close to the bus station. Another problem with the spider maps is that one has no idea of the interconnection of roads outside of the rather local area covered in the central inset. So, considering the example of Figure 2.15, if someone is trying to get to a place out of the local area from South Kensington station, s/he would look at the spider map and surmise it is not easily possible. With the old conventional local area map s/he could have realized that a short walk up some near street would have taken her/him onto the routes which go directly to the required station. An ideal compromise could be to have the spider map and the conventional one in two inside panels on bus shelters with space to do so.
Figure 2.7: Example of the public transportation map of Zurich, Switzerland. 

Figure 2.8: Example of the bus map of Porto, Portugal. Source: ‘Rede Diurna’, design by Z-CARD™ Pocketmedia™ [ZCard, 1999].
2.7 Examples

Figure 2.9: Example of a subway map of Madrid, Spain. *Source:* 'Rede de Metro Y Cercanias', design by Consorcio Regional de Transportes de Madrid [CRTM, 2000].

Figure 2.10: Example of the bus network map of Copenhagen, Denmark. *Source:* 'HT-bussernes rutenet', design by Damsgaard og Lauge [HT, 2001].
Figure 2.11: Example of the railway and underground network map of Amsterdam, Netherlands. Source: 'Gemeentevervoerbedrijf Amsterdam', design by Hans van der Kooi from Samenwerkende Ontwerpers [Pedersen, 1988].

Figure 2.12: Example of the rapid transit map of Baghdad, Iraq. Source: Baghdad Metro, design by Richard Dragun from Design Research Unit London [Pedersen, 1988].
Figure 2.13: Example of a bad schematic map.
Figure 2.14: Two subway lines and their connections in Tokyo, Japan. Source: Yamanote Line and Chuo Line, City guide to Tokyo, Access Press [Pedersen, 1988].
Figure 2.15: Example of a spider map, London, England. Source: Bus services from South Kensington, design by Transport for London [TfL, 2001].
Chapter 3

A Framework for Electronic Schematic Maps

We provide here a framework which enables users to visualize the vector-based, cartographic information of a transport database, and to interact with location information on the network routes. The framework offers a user-friendly solution to query a schematic map. Answers to users queries are done by highlighting objects in the electronic map. The idea is that users can interact with the schematic map as mapmakers in the map production and as map readers through location queries.

In order to generate electronic schematic maps, we need a database with transportation information of the city or region under consideration. The database should also contain topological information on the transport network, such as connectivity and stops, in order to answer location queries from map users related to the transport network. For specifying transportation network elements at a conceptual level, we provide a data model to describe characteristics of a public transport network [Avelar and Huber, 2001]. We have built an interface on top of this data model for supporting location queries. The tool can be used in display devices with small screen, such as with palmtop and mobile technology, and terminal sites located at train or bus stations, tourist bureaus, etc.

The chapter is organized as follows. In Section 3.1, we give a short introduction to prior work on modeling data for transportation. In Section 3.2, we present user location queries for schematic maps and characteristics to be taken into account when modeling the transport data. Then, Section 3.3 describes how the reality has been formalized in our data model. In Section 3.4, we present a prototype tool to visualize and explore a transport network. Finally, Section 3.5 gives our concluding remarks and outlines further work.
3.1 Modeling a public transport network

Electronic schematic maps can answer user location queries conveniently using the information stored in a database. We need, therefore, to create a database which has two major purposes: to be used as a basis in the automatic generation of schematic maps, and as a general information system to users of a public transport network. Regarding the first purpose, we need to store geographical information about the transport routes. In order to answer user location queries, topological characteristics of the transportation network, such as intersections of routes and rivers, adjacency of stations, and proximity of stations to special places, should also be stored.

Data models are a fundamental aspect in a GIS project, because they describe how the geographic reality will be represented in the computer [Halpin, 1995]. Data modeling aims at designing efficient and consistent databases. The result of this process is a conceptual scheme which reflects the requirements that the desired database must achieve.

Current data models for representing streets and routes in a transport network integrate the cartographic representation, the network link, and attributes of the link into a single linear spatial object [Duecker and Butler, 2000], such as the ArcInfo georelational data model. Nielsen et al. in [Nielsen et al., 1997] describe their experience in using ArcInfo to handle public transportation networks. They state that ArcInfo, and most GIS-packages, are currently not able to handle fundamental aspects of public transport networks. The translation from transport model topology to GIS-topology has to be carefully treated, because transportation network models can demand a complex topology not very well covered by most traditional GIS-packages. In fact, the data model to be described in this chapter was also implemented using ArcInfo and ArcView, in order to build an experimental database with public transportation network information of a selected perimeter of Zurich, and some data types could not be directly defined in ArcInfo (see [Purtschert, 2001]). In ArcInfo, there is a topological connection between nodes and links (arcs) given by the Arc-concept. The Arc Attribute table contains information on the 'from node number' and the 'to node number', however, the opposite information on which links are connected to a node is not contained in the Node Attribute table.

Other different data models have also been proposed in Geographic Information Systems for Transportation (GIS-T). For example, Vonderohe et al. in [Vonderohe et al., 1997] describe a data model prepared by various transportation professionals and agencies in an effort to provide data sharing of linear systems for GIS-T. Linear systems are used in many application areas based upon networks,
including infrastructure management, transit systems, waterway navigation, hydrological analysis, utilities management, and seismological sensing. The central notion is that a common linear datum relates links and nodes of linear systems to the real world. But similar scale, same data model, and same segment roads were necessary to match spatial data, therefore this was not a satisfactory way to model the transport data. Another example is given by Duecker and Butler in [Duecker and Butler, 2000]. They discuss transportation data sharing and the need for a data model that holds transportation features, not their graphical representations, as objects of interest. They provide a data model in which cartography is directly connected to the transportation feature, and not to the linear datum as in Vonderohe et al. [Vonderohe et al., 1997]. However, attempts to use the model in the implementation of data tied to other representations of a linear network is usually problematic [Duecker and Butler, 2000].

Sharing transportation data and linear data models are an important issue, and a difficult one, because of the varied nature of applications that require data in specific forms. The data model here presented has an object approach. It calls for an uncoupled approach of graphics, topology, and characteristics of transportation features to facilitate comprehension and maintenance of the database. The transportation features were selected to better represent the requirements of our application.

### 3.2 Model requirements

A schematic map aims to provide topological information on transportation to answer user queries about location and directions. The data model must reflect all necessary information, but not more, to support the automatic generation of schematic maps from our database and to answer certain user location queries completely and efficiently. The data model should also support operations associated with topology (e.g., connectivity, crossing, and proximity) and with network analysis (e.g., path finding, and location), in order to answer location queries. Next, we give the user queries here considered and other details about the information required to be stored in our transport database.

#### 3.2.1 Queries

In order to design the data model effectively, we checked which are the basic questions of transportation riders for schematic maps [Monmonier, 1996]. The following queries should be successfully addressed by our data model:
• Where is my destination on the map? In which direction do I need to go?
• What is the name of the station at end of the line?
• What bus, tram, etc. takes me from A to B? Do I need to change lines? If so, where and what line?
• How many stops do I ride before I get off? How long will it take?
• How do I get to the other side of a river or lake?
• What is the nearest station to my present location? What is the nearest station to a given place?

3.2.2 Considerations

Next to the geographical information about the spatial location of transportation network elements for the generation of schematic maps, we need to store lakes and rivers and their geographic shapes in the database. Topological information related to the intersection between transport routes and bodies of water is also required to simplify the answer to queries like “How can I get to the other side of a river?”. However, topological information on the intersection of routes is not necessary to be stored, because neither the algorithm to generate schematic maps needs it - the generation algorithm can detect topological changes in the linear network [Avelar and Müller, 2000] (details in Chapter 5) - nor this information is needed to answer above queries. Figure 3.1 shows the main objects and relationships to be stored in our database.

In order to answer proximity queries, ‘nearTo’-relations among map features have to be explicitly identified and stored. For example, for the query “What is the nearest station to a given place?”, we have to store special reference places, such as zoos, parks, stadiums, theaters, etc., with their geographic location and the stations they are conveniently located next to.

For queries like “How do I get from A to B?” we need to identify when changes between routes are possible to occur. Transport routes can connect to each other at various stations along their routes. Most of the routes have only one way in both directions, but there exist stations which are served in one direction only. A certain journey might use different transport modes, because a public transport network often contains different networks, such as trains, trams and buses. Additionally, transit service is time dependent, i.e., the best path between an origin and destination can change depending on the timing services available. At the same station,
different routes may have different wait times. The data model should also provide means to describe these characteristics of a public transport network.

### 3.3 The transport network data model

The transport network object model is shown in Figure 3.2. The graphical notation used is based on Entity-Relationship (ER) Diagrams [Chen, 1977]. This approach to conceptual design of a data model provides a simple way of declaring the objects, attributes, and relationships among them. The diagram symbols are summarized in the following paragraph for convenience.

Entities may be seen as real-world objects which are of interest for the application under consideration. They are shown as boxes, which also include their main attributes. Relationships are represented by lines, and they are used to model associations. Entities are related to each other through verb-oriented statements. A relationship without explicitly stated verbs generally can be read as an ownership relationship. The minimum and maximum number of occurrences of each entity that may participate in a relationship are characterized using (min,max)-notation. If an entity may or may not participate in a relation then min is 0. The cardinality for
Figure 3.2: Transport network data model.
'many' is indicated by "*". The arrows underline the semantics of the description.

### 3.3.1 Overview of transportation features and relationships

The objects used for modeling the public transport network can be categorized into three basic groups which can be identified in Figure 3.2. The first group is transport features, whose boxes are represented by continuous lines. The second one is geographic locations, which uses dotted lines in their boxes. The third group is path objects, represented by dashed line boxes. We describe following the main entities and relationships of these groups:

- **Transport features**: Elements of the transport network and other geographic objects required to answer user location queries. A Line is a route composed of LineSegments (oriented, between two neighbor stations). Each LineSegment may be formed from one or more SegmentShapes (non-oriented, for storing the geometric shape of a line segment). The same SegmentShape can be shared by different LineSegments. Stations can serve one line only - StopStations - or be connection nodes, where travelers can change lines - LinkStations. We still introduced the concepts of WaterBody, e.g., rivers and lakes, and of ReferencePlaces, which are special places of the city located in the geographic context of the transportation network. The relationships we need to store contain topological information about the intersection of SegmentShapes and WaterBody (‘crossing’) and of proximity of ReferencePlaces to Stations (‘nearTo’).

- **Geographic locations**: Real world, earth-based location of transport features. Occurrences of transport features can be linear between two points, when we use the object Edge, or in a single geographic location, that is given by the object Point. To represent a sequence of linear segments or area elements, such as lakes, we use the collection EdgeSet, which are open or closed linearly connected sets of Edges.

- **Path objects**: Description of a path through the transportation network consisting of one or more segments. A Path through the network is composed of Steps. A Step is a section you can go from one link station to another without changing the vehicle. The relationships in this group are: ‘nearestLink’, which assigns one or two link stations to every stop station, ‘leadsTo’, which is a matrix with paths from every link station to every other link station as entries, and ‘using’, which assigns a path to every matrix cell of ‘leadsTo’. The detailed solution to model paths is described in the next subsection.
To populate the database, a number of constraints should not be violated. Such constraints should be considered when entering data and in the checking mechanisms of the data model. We used here:

C1. A Line has at least two Stations.
C2. A Station has at least one Line.
C3. All LineSegments of a Line must be of the same transport mode.
C4. A StopStation has only one Line, and a LinkStation has at least two Lines.
C5. There are at most two LineSegments in a StopStation (coming from or going to) and at least two LineSegments at a LinkStation (two in case of a terminal of two Lines).
C6. The first and last Point of the SegmentShape.formedBy of a LineSegment must be identical to the Points of Stations in endsAt.
C7. Line.consistsOf must form a cycle (to cover both directions of a Line).
C8. Edges of EdgeSet must be linearly connected.

3.3.2 Reducing response time and storage requirements to path queries

When modeling paths to answer the "How do I get from A to B?" type of queries, we can take advantage that routing is a common problem in computer science and that there exists in practice many routing solutions. Because the data related to line routing in a public transport network is in general more or less static over several months or even years [Kern, 1986], it can be reasonable to perform the operation of calculating paths from one station to another in advance and store the results in a database. The trips can be routed using any existing shortest path routine in the preferred GIS. Queries can profit by accessing pre-calculated paths directly. This can reduce the response time to the path queries.

In this solution, a path consists of steps, which are parts of the route between link stations without changing vehicle. A path itself has an attribute called timeNeed. This is the sum of timeNeed for each step and the times needed to change between the steps at link stations (attribute maxInterval of object Line). The straightforward approach would lead to a square m x n adjacency matrix, where m = n = the number of stations in whose cells the paths are stored. Because this matrix would become very large (e.g., for a city with 500 stations, there would be 250000 cells), we searched for a way to reduce storage requirements.

The idea is to store only paths from link station to link station and complete the actual paths during runtime of queries. Link stations are usually only about 20% or 40% of all stations, so the number of cells would result in about 15% of the original
3.3 The transport network data model

amount. To complete paths from a link station to a stop station, we use the stored data in the ‘nearestLink’ relationship. Figure 3.3 illustrates the proposed solution using pre-calculated paths. The algorithm is following presented.

1. Determine link stations

   * A is stop station, B is stop station
   2 adjacent link stations to A (A<sub>i</sub>, A<sub>j</sub>) x 1 link station to B (B<sub>j</sub>)

2. Get pre-calculated paths

   * Step: (start, end, line, direction)
   + L2: maxinterval = 6 total time
   | Step 1 | Step 2 |
   Path (A<sub>i</sub>, B<sub>j</sub>) = [(A<sub>i</sub>, G, L4, H), (G, B<sub>i</sub>, L2, B<sub>j</sub>), 10]
   Path (A<sub>i</sub>, B<sub>j</sub>) = [(A<sub>i</sub>, B<sub>i</sub>, L1, L2, B<sub>j</sub>), 2]

3. Extend paths

   Path<sub>1</sub> (A, B) = [(A, A<sub>i</sub>, L3, C), (A<sub>i</sub>, G, L4, H), (G, B, L2, B<sub>j</sub>), 25]
   Path<sub>2</sub> (A, B) = [(A, A<sub>i</sub>, L3, K), (A<sub>i</sub>, B<sub>i</sub>, L1, L), (B, B, L2, B<sub>j</sub>), 20]
   Path<sub>3</sub> (A, B) = [(A, A<sub>i</sub>, L3, C), (A<sub>i</sub>, G, L4, H), (G, B, L2, B<sub>j</sub>), 25]
   Path<sub>4</sub> (A, B) = [(A, A<sub>i</sub>, L3, K), (A<sub>i</sub>, B<sub>i</sub>, L1, L), (B, B, L2, B<sub>j</sub>), 20]

4. Return shortest path

   Return Path<sub>2</sub> (A, B); (Path<sub>2</sub> = 20 < Path<sub>1</sub> = 25)

**Procedure** GetPathfromAtoB (A, B, matrix);
Input: stations A and B, matrix with paths of linkstation x linkstation
Output: shortest pre-calculated path from A to B

begin
   if A is a stop station then
      get nearest link stations to A in A<sub>i</sub> else A<sub>i</sub> = A;
   endif
   if B is a stop station then
      get nearest link stations to B in B<sub>j</sub> else B<sub>j</sub> = B;
   endif
   get pre-calculated path<sub>k</sub> for combinations A<sub>i</sub> x B<sub>j</sub>; (* maximum of 4 paths *)
   if A or B is a stop station then
      extend path<sub>k</sub> by steps from A or B to respective nearest link stations;
   endif
end

Figure 3.3: Illustration of use of pre-calculated paths.
compare all $path_k$ and return shortest;
end.

3.4 Example of use

We created a dataset to produce an experimental design and visualization to exemplify the use of our data model. The experimental scenario is thought to give a better understanding of the model. A database containing this dataset was built and accessed by using OMS Java [Kobler and Norrie, 2000, Huber, 2001]. In Figure 3.4, we show the scenario objects and their identification. We give below the description of the objects and relationships according to the data model.

![Figure 3.4: Scenario example.](image)

Description of objects:
- Point: \{P1, P2, ..., P18, S1, S2, ..., S10\}
- Edge: \{E1, E2, ..., E28\}
- EdgeSet: \{lak, riv, S1S2, S2S3, S4S5, S10S6, S6S7, S7S3, S8S4, S8S9, S9S8\}
- RefPlace: \{Zoo\}
- Station: \{S1, S2, ..., S10\}
3.5 Querying the database

We present following the framework developed to provide data visualization on a transport database based on the described data model and query capabilities.
3.5.1 Purpose of TNview

The TNview (for Transportation Network viewing) tool enables a user to view the transport network and to explore and analyse geographical and topological information through cartographic queries. The experimental system was implemented using Java. We can start TNview either as an application or as an applet embedded into a HTML page through Internet [Huber, 2001]. The structure of the tool is illustrated in Figure 3.5.

The main module (applet or application) is TNview, which is responsible for the graphic user interface. Map visualization is the module responsible for painting the map contents and creating the legend of transport modes. This module contains also the data structure for referencing the transportation information in the database. It can therefore perform queries, such as finding a transport object by name or by coordinates, and show the answers in the map.

The module event handling is the listener of all events. It delegates tasks to all components of the system. Queries can be activated by mouse clicks, mouse movement, buttons, text field, or menu items.

A separate module for performing a transformation of coordinates is necessary, because there are two coordinate systems involved. The first is the global one, which is used for spatially placing objects in the database and, in general, is according to some official coordinate system of the town or country. The second coordinate system is related to the screen, whose coordinates relate to the origin of the painted map in the global system, and their size depends on the present zoom.
3.5 Querying the database

level. Each time an element needs to be displayed, its geographic coordinates are transformed into screen coordinates, and vice-versa in case of element-at-position queries.

The TNview tool and the transportation network of our scenario is shown in Figure 3.6.

![TNview tool and scenario example](image)

Figure 3.6: Visualization tool and scenario example.

3.5.2 User interface

The user interface allows to evaluate queries to obtain transport lines, stations, reference places and lakes, by name or geographical location. Advanced queries, such as nearest stations to a reference place, are selected from the menu bar.

Transport features can be selected directly from the map, by double clicking on a map element, or according to queries. The selected element is highlighted on the map and its attributes are displayed in the text area (see line Bus1 in Figure 3.6). The attributes in the text area can be extracted for further purposes by copy/paste mechanism. By moving the cursor to a map element, its name is shown in the status bar. A button in the upper right side of the interface gives access to toggle an anti-
aliasing function for graphical treatment of lines. Zoom is also provided. These buttons make easy to browse around the map to find information about surrounding areas.

The map legend for mode of transportation is automatically generated, such that each mode has a different line specification. In case a route segment is shared by many modes of transportation, even all of them, TNview ensures that all modes are always visible on the map. This is achieved by enlarging the width of lines and reducing brightness of color for increasing the number of transport modes.

3.5.3 Data structure

The map visualization module has an attribute workspace, which is the central connection to the database contents in the application. To avoid sequential scanning of the whole workspace for every repaint and element-at-position query, which occurs every time a user moves or clicks the mouse inside the map, a two-dimensional index data structure is laid over the map. In the case of schematic maps, a simple space-driven data structure based on a regular grid was adequate, because elements of a transport network are usually homogeneously distributed. Thus, there is no need for using sophisticated spatial partitions dividing the data, for example, according to their density, because there are no characteristic clustering zones. For more information about spatial data structures refer, for example, to [van Kreveld et al., 1997] and [Jones, 1997].

The whole space is divided into the same amount of index-cells in both dimensions. Those cells have a linked list as attribute which store stations, reference places, line edges, river edges and lakes, that fall into their region or intersect it. Elements that fall into several cells have multiple references. If the user clicks on a certain position, only one cell is scanned. A separate list references all objects which should be highlighted in the selection mechanism. Selectable objects are Points, Edges, and EdgeSets. The illustration in Figure 3.7 shows the used data structure.

3.6 Conclusions

The two major contributions of the work presented in this chapter are: (1) we designed a data model for describing geographical and topological information of a public transport network and implemented it; and (2) we developed a prototype tool to support queries on the data model.
Data modelers can use this work and their previous experience to build upon the described object data model, adapting it for other particular scenarios in which streams of objects at nodes in a network play a role. As described by Vonderohe et al. in [Vonderohe et al., 1997], roadways, railroads, transit systems, shipping lanes, and air routes have all linear features that can utilize a same basic network data model.

A natural extension of the transport data model would be to include timetables. One possible simple approach to integrate timetables in the data model is drafted in [Huber, 2001]. Timetable is a type with no attributes, and course has StartTime, StartStation, and EndStation as attributes. A course represents a vehicle driving a route once (for- or backward). The StartTime of the course is the only necessary time information, the rest is given by timeNeed of LineSegments. A Timetable has a set of courses. The courses are different for weekdays and weekend. Start- and End-Station of a course are needed, because in the morning the vehicles come out from a depot, which is not necessarily located at every line terminal station. So the course can start anywhere in the route. Furthermore, there exist lines that circulate only during peak times the fully route. The problem of the extension by schedules is to rethink the pathing algorithms.
Future work could include the performance evaluation of the use of pre-calculated paths and the extension of the data model for including other features, such as street network and walkways. Although schematic maps, in general, offer little or no references to surrounding streets, this interaction can be useful when users want to find stations near to specific streets on the map.
Chapter 4

Background: Schematic Maps on Demand

Previously, we have presented basic characteristics for the design of schematic maps and a model for representing transport network elements in a computer. In this chapter, we present our approach to generate automatically schematic maps on demand, i.e., by request from users. We also give an overview of related work in the domain of schematic maps on demand.

First, we show in Section 4.1 that automated generation of schematic maps can be regarded as an attempt to meet a set of constraints. The necessary terminology for when schematic maps are treated as an optimization problem is reviewed in the same section. Then, we briefly review other approaches to automate the generation of schematic maps in Section 4.2.

4.1 Schematic map as a geometric constraint problem

Schematic maps, such as almost any layout problem, have a set of constraints on how the information can be laid out. An individual who wants to produce a schematic map for a city is faced with geometric and aesthetic design constraints for the construction of such maps. The constraints involve fixed angles for schematic directions, minimum distance separating pairs of map features, minimum route overlapping distance, and maximum length of line segments. Thus, generating schematic maps on demand can be regarded as a geometric constraint problem that should be solved algorithmically, in interactive time.

From the computational point of view, the question is how to assign a set of constraints to a route network. Just like in the manual and assisted methods to
produce schematic maps, we can move the lines of the original map, stretching or shortening them, to new positions that fulfill the map constraints better than before. Schematic maps on demand can approach this problem by a number of iterations, i.e., using iterative algorithms: the mapmaker chooses the constraints, the iteration is run and route displacement takes place.

The schematization of routes, such as with layout problems, can be posed as a search for an optimal layout over a space of possible layouts. The quality of a solution can also be evaluated using constraints.

Before getting into the iterative approach, it is important to define terms which will be used. In order to better understand the context of the approach, we discuss the concepts of constraint, heuristic search, iterative algorithms and other related terminology.

4.1.1 Constraints

Constraints naturally occur in many situations. For example, we all must restrict ourselves in order to fulfill the constraints emerging from law, social morality, financial matters, etc. Although it is often easy to state the constraints to be fulfilled, we know how difficult it is to satisfy them simultaneously [Seybold, 1999].

In cartography, constraints can be considered as limits which indicate requirements for managing space, establishing neighborhoods of objects, preserving topological relations, or establishing other information for map features. A hard constraint may not be violated and defines the needs of how a feature may be placed on the final map. A soft constraint suggests a placement guideline - perhaps a limit which should not generally be violated, but may be relaxed under certain conditions [Brazile, 1998].

There are situations in which we are confronted with more informal, less explicit constraints. Layout problems, just like spatial reasoning, involve an informal description of how to place objects in space such that a number of constraints are satisfied. For instance, they demand that a certain entity is “on” another entity, that two entities “intersect”, or that an entity is “included” in another one.

An important notion for the definition of constraint is the one of entity, which is something that has its own, unique existence and can be individually identified, separately from other entities. The entity attributes define the state of a system. Graphical entities, for example, have attributes which can be, in general, divided into three main categories. Attributes such as size, shape, and location are geometric attributes, while attributes such as color and texture are aesthetic attributes.
Relations between graphical entities, such as above/left-of and connected-to, are topological attributes.

A constraint is an explicit declaration of a condition that is specified for one or more entities, or a relationship between two or more entities. A constraint condition is similar to the predicate in a production rule IF <predicate> THEN <consequence>. The distinction is that a constraint is not bound to a particular action. The overall rule is that all constraints must be satisfied, and any number of actions can be applied to resolve them.

If there are insufficient constraints to control all states of an entity, the entity is considered to be “under-constrained”. If there are more constraints than necessary, it is “over-constrained”. An entity, for which there is no solution that can satisfy all constraints, because of it being over-constrained, is “inconsistently over-constrained” [Bettig, 1999].

### 4.1.1.1 Constraint types

The terminology for identifying types of constraint can vary for different contexts and authors. Table 3.1 presents examples of constraints and their type terminology.

<table>
<thead>
<tr>
<th>Example of constraint</th>
<th>Type terminology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance, angle</td>
<td>Geometric design constraint, dimensional constraint, metric constraint</td>
</tr>
<tr>
<td>Coincident, incident, tangent, concentric, coaxial, parallel, perpendicular</td>
<td>Geometric constraint, dimensional constraint, structural constraint</td>
</tr>
<tr>
<td>Radius, major axis, focal distance</td>
<td>Geometric property definition constraint, dimensional constraint, metric constraint</td>
</tr>
<tr>
<td>Fixed entity, fixed coordinate, fixed direction</td>
<td>Geometric property definition constraint</td>
</tr>
<tr>
<td>Horizontal distance, vertical distance</td>
<td>Dimensional constraint</td>
</tr>
<tr>
<td>Distance along curve, midpoint, symmetric area</td>
<td>Geometric constraint, position constraint, structural constraint, engineering constraint</td>
</tr>
<tr>
<td>Equations, inequalities and tables of values</td>
<td>Arithmetic constraint, algebraic constraint, engineering constraint</td>
</tr>
<tr>
<td>Vertex incident with 3 faces, order of points</td>
<td>Topological constraint</td>
</tr>
</tbody>
</table>

Table 3.1: Example of constraint type categories.
Bettig in [Bettig, 1999] gives examples of other constraints and their type terminology. We call the main types of constraints we are concerned with in generating schematic maps as geometric constraints and topological constraints. Geometric constraints are inherent in most of our environment, even though we may not recognize them at first. They are used to describe spatial situations. For example, in assembling parts of a mechanism, there are hints how to realize the geometric constraints to put the parts of the mechanism together. Although the concrete realization is not given (contrary to geometric constructions), a solver can directly derive actions to satisfy the constraints. Topological constraints refer to self-intersection, order and connectivity of entities, and "topologically correct" constraints.

4.1.1.2 Geometric problems

The kinds of problems that have been considered as geometric problems are, in general, those of the form: construct a shape corresponding to a specific description of it. These problems involve not only geometric construction and reasoning, but often also topology. There is a great number of types of geometric problems that must be solved in different applications. The geometric problems include satisfying constraints – as it is the case in this thesis –, checking over-/under-constraint conditions, finding relationships between two entities and matching pattern to model [Bettig, 1999].

There are many approaches to solve geometric problems. Most of them involve finding a single solution for well-constrained situations, such as in assembly problems. In spatial reasoning, the geometric and topological constraints only serve to define the allowed set of possible solutions. It is therefore required to find a solution that optimizes a certain cost function.

In the next subsection, we give more details about how to solve a geometric constraint problem. Note that we place a limit on the research and present only solutions that involve optimization.

4.1.2 Heuristic search

When a problem can not be solved by deterministic means either because there are too many potential solutions to make total enumeration feasible or because it is NP-hard [Kozen, 1991, Hu, 1982, Sedgewick, 1993] or has no explicit analytical solution that is proven correct, then maybe it can be solved by either trial and error or some other clever method.
Often brute-force or exhaustive searches of all possible solutions are infeasible, because of the amount of computer time needed. In many complex search problems, there is a combinatorial explosion in the number of possible solutions that would need to be investigated. However, there are many situations in which is sufficient to find a feasible solution, i.e., certain constraints are satisfied and the solution is close to optimal, thus exploring the entire state space tree may not be necessary.

Heuristic search is regarded as a class of computer-based methods able to find the approximate answer to problems that cannot otherwise be given exact solutions, usually because there is a large, sometimes extremely large, number of possible solutions that may have to be examined [Openshaw and Openshaw, 1997]. Heuristic search procedures are not necessarily optimal, because they are not exhaustive. Solutions to a problem will be produced, without even knowing how good (or bad) the best solution really is.

The Oxford Reference Dictionary defines the adjective heuristic as “serving or helping to find out or discover; proceeding by trial and error”. In the context of algorithms, heuristic will be a method of performing a minor modification, or a sequence of modifications, of a given solution or partial solution, in order to obtain a different solution or partial solution [Kreher and Stinson, 1998]. A heuristic algorithm will, therefore, consist of iteratively applying one or more heuristics, in accordance with a certain solving strategy. The term heuristic algorithm describes an algorithm that tries to exploit a certain combinatorial structure or solve an optimization problem1 by the use of heuristics.

The difficult aspect of characterizing a problem as a search may center either around finding a suitable goal state whose features can be described in some meaningful way, or around defining a search strategy that covers a significant portion of the search space [Agrawala and Stolte, 2001].

Provided that the constraints of a problem can be quantified in mathematical terms, four concepts usually occur in search problems [Openshaw and Openshaw, 1997]:

1. Objective: There is a goal or purpose to be met to the best possible extent.

2. Search space: There is a small or large or almost infinite universe of possible solutions that need to be considered and explored, either fully or partially, explicitly or implicitly, while searching for the best possible result.

---

1The terms “optimization problem” and “search problem” are considered synonymous. The optimization problem is the search for the feasible solution for which the profit is as large as possible.
3. Changing a solution: There needs to be some means of moving around the theoretical solution space, in order to find the best possible desired result.

4. Evaluation function: An evaluation function can be constructed to determine the degree to which constraints are met, thereby selecting solutions that are superior to others with regard to the constraints. In every real-world context, we have to choose the evaluation function, because it is not given with the problem.

A well-established approach to solving search problems is to use iterative algorithms. To describe this concept, Ware and Jones [Ware and Jones, 1998] use the illustration by Russell and Norvig in [Russell and Norvig, 1995] by considering all states (in our case, all map realizations) to be laid out on the surface of a landscape. The elevation at any point on the landscape represents the quality measure returned by the evaluation function for the particular state at that point. An iterative improvement algorithm will move around the landscape in an attempt to find the highest troughs, which correspond to optimal states.

The main design strategies for iterative improvement algorithms, i.e., the means by which we design a neighborhood search and incorporate it into a heuristic search algorithm, include hill climbing algorithms and simulated annealing. In a hill climbing or simulated annealing algorithm, we begin with an initial feasible solution and proceed to construct from it a sequence of feasible solutions by applying a heuristic, which is in turn a neighborhood search technique. Hill climbing algorithms always make changes that improve the current state (movement in the landscape is always uphill), whereas simulated annealing algorithms can sometimes make changes that make things worse (movement is sometimes downhill).

We describe next the hill climbing and simulated annealing techniques. For more details about heuristic search and iterative algorithms, see [Kreher and Stinson, 1998] and [Michalewicz and Fogel, 1998], or other specialized literature in the areas of combinatorial algorithms and artificial intelligence.

4.1.2.1 Hill climbing

Hill climbing is a heuristic strategy in which the algorithm attempts to proceed toward an optimal solution by finding a sequence of feasible solutions, each of which is better than the previous one. The analogy of climbing a hill refers to the walker trying to find his or her way up a hill making only upward moves, while being unable to see the summit. The "hill" is a function of some kind. At any location the function can be evaluated and the direction of ascent identified. Imagine moving
up a bit and then re-evaluating the function, this is analogous to how this optimization procedure works.

In a typical optimization problem, there may be many locally optimal solutions that are not globally optimal solutions. Using the landscape analogy, a local maximum can be thought of as a peak in the landscape that happens to be lower than the highest point on the landscape.

Hill climbing techniques can become stuck because of the unevenness in the landscape being searched. This occurs when the search ascends to a local maximum, from which all moves appear to generate a worse state.

Hill climbing can only provide locally optimal solutions, and these solutions depend on the selection of the starting state. For these reasons, this method is often considered restrictive. However, several ways of trying to deal with the problem of local maxima are available, such as random-restart of the initial state, backtracking, and multiple moves [Ware and Jones, 1998]. But the exponential nature of most realistic search spaces can make such remedies impractical. Suboptimal results can sometimes be avoided by using another heuristic technique that will not get stuck every time a locally optimal solution is encountered, e.g., simulated annealing [Kreher and Stinson, 1998, Michalewicz and Fogel, 1998].

The denomination hill climbing could equally well be hill descending, the same principles apply, and the term gradient descent method is used instead. Hill climbing implies a maximization problem and the equivalent descent method is envisioned for minimization problems [Openshaw and Openshaw, 1997].

There are a few versions of hill climbing algorithms. They differ mainly in the way a new solution is selected for comparison with the current candidate solution. We present below a generic hill climbing algorithm [Kreher and Stinson, 1998]. If a neighborhood search does not find any new state better than the current one, it must return \textit{Fail}. The variable $c$ keeps track of the number of attempts to solve the problem.

\textbf{Procedure} GenericHillClimbing ($c_{\text{max}}$);
\begin{verbatim}
begin
  $c \leftarrow 0$;
  choose a starting feasible solution $X$;
  $X_{\text{best}} \leftarrow X$;
  \textit{searching} $\leftarrow$ \textbf{true};
  \textbf{while} \textit{searching} \textbf{and} ($c \leq c_{\text{max}}$) \textbf{do}
    select operator(s) not yet applied to $X$;
    apply operator(s) to produce a new state $X_{ni}$;
end
\end{verbatim}
if $X_n \neq \text{Fail}$ then
    $X \leftarrow X_n$;
    if $X$ is better than $X_{\text{best}}$ then
        $X_{\text{best}} \leftarrow X$;
    endif
else searching $\leftarrow$ false;
endif
$c \leftarrow c + 1$;
end while
end.

4.1.2.2 Simulated Annealing

Simulated annealing is a popular method of attempting to allow the search to escape from locally optimal solutions. It is based on an analogy with a method in metallurgy of heating and cooling metal which is known as “annealing”. Instead of the function only making uphill moves, with this method it can also move downhill [Kreher and Stinson, 1998]. In simulated annealing, we use a randomized neighborhood search strategy, i.e., the algorithm does a random walk on the search space. The evaluation function of a state is called energy $E$, and the main parameter of the algorithm is called temperature $T$.

Beginning in a starting state, the algorithm repeatedly selects a random neighbor, computes the energy difference of the old and the new state, and accepts the new state depending on the result of the comparison [Marzetta, 1998]. As with gradient descent, simulated annealing always accepts a new state if it is better (lower energy) than the current state. A new state worse (higher energy) or equal to the current state is accepted with a certain probability $P$ less than 1, depending on the energy difference

$$P = e^{-\Delta E/T}.$$ 

At high temperatures, every new state is accepted; at low temperatures, a new state is accepted only if this operation decreases the energy. In practice, the probability $P$ is usually tested against a random number $r$, where $0 \leq r \leq 1$. A value of $r < P$ results in the new state being accepted. If $P = 1/3$, for example, then we would expect, on average, for every third worse new state to be accepted.

The algorithm starts with a high temperature $T$ ($T_0 > 0$) and proceeds gradually decreasing the value of $T$ until the temperature is so low that the random walk stops at a local minimum. Usually, $T$ is decreased after each iteration according to a cooling schedule, following the formula $T \leftarrow \alpha T$, where $0 < \alpha < 1$ is some
constant (usually $\alpha$ is close to 1). The initial choice of $T$ and the rate at which it is decreased has an effect on the convergence properties of the algorithm and on how well it works [Kreher and Stinson, 1998]. Generally, the slower the rate of change, the better the result, but suitable values are decided upon after some preliminary experimentation.

Implementations of simulated annealing differ basically in the ways for decreasing the temperature and in the termination condition. Simulated annealing will often consume at least 100 times more time than a simpler hill climber. We present below a generic simulated annealing algorithm [Kreher and Stinson, 1998]. It does not differ from the structure of the iterated hill climber very much. The variable $c$ keeps track of the total number of iterations. When $c_{max}$ iterations are performed the algorithm terminates.

**Procedure** GenericSimulatedAnnealing ($c_{max}$, $T_0$, $\alpha$);

**begin**

$c \leftarrow 0$;

$T \leftarrow T_0$; (* $T_0 > 0$ *)

choose a starting feasible solution $X$;

$X_{best} \leftarrow X$;

**while** ($c \leq c_{max}$) **do**

select operator(s) not yet applied to $X$;

apply operator(s) to produce a new state $X_n$;

if $X_n \neq Fail$ then

if $X_n$ is better than $X$ then

$X \leftarrow X_n$;

if $X$ is better than $X_{best}$ then

$X_{best} \leftarrow X$;

endif

endif

else $P \leftarrow e^{-\Delta E/T}$;

generate random number $r \in [0, 1]$;

if ($r < P$) then

$X \leftarrow X_n$;

endif

endif

revise $T$ according to annealing schedule; (* $T \leftarrow \alpha T$ *)

$c \leftarrow c + 1$;

**end while**

**end.**
4.1.3 The iterative approach

For the purposes of automation, schematic maps on demand as a geometric problem may be regarded as a search problem. In the production of such maps, we have to displace many different lines representing trams, trains, buses, etc., according to the geometric and topological constraints of the map, which involve fixed angles, minimum distances and non-intersection. The task is to find an arrangement such that all lines are on schematic directions and the overall deformation in angle, shape and length of line segments is minimized.

If each of the $n$ map objects is subject to $k$ possible states, then there is a total of $k^n$ alternative realizations of the map [Ware and Jones, 1998]. The set of all possible states in which a schematic line network can find itself constitutes the search space. Obviously, attempts to solve the problem by an exhaustive search of the space of candidate maps are impractical and it is necessary to consider sub-optimal strategies. If each possible map state is subjected to evaluation with regard to the extent that it satisfies the map constraints, then it is possible to envisage applying optimization strategies to find the map state that best meets the constraints. But the search is for an acceptable, rather than the 'best' schematic map.

We can apply the comparison made by Ware and Jones in [Ware and Jones, 1998] between map generalization and point-label placement problems to the schematic map problem. In schematic maps, as well as in map generalization, the number of possible states can be expected to be considerably larger than in point-label placement, due to the range of possible constraints that could be given to each object. The cost of evaluating the function can also be expected to be greater for the same reason. And the optimization for point-label problems is NP-hard [Kozen, 1991, van Kreveld et al., 1999, Strijk, 2001].

In this thesis, we use an iterative algorithm based on the displacement of points to produce schematic maps on demand. We want to produce maps with connections in schematic directions, and minimize angular error deformation. Previous work on similar iterative displacement of objects has been also done in map generalization; they attempted to minimize the number of conflicts observed [Ware and Jones, 1998] or some function of the total amount of conflict [Harrie, 1999, Ruas, 1998].

Starting from an initial configuration with all points located at their original location, each point in turn is considered. Any point location that does not satisfy the set of constraints defined for the schematic map is moved in an attempt to satisfy them. The process is repeated until the displacements made become small enough or until it meets a user defined stopping criterion. The method used focuses on re-
4.2 Prior related work

As we mentioned in Chapter 2, the cartographic challenges of linear thematic mapping are not entirely new, but the volume of such activity is expanding only recently, especially for the visualization of schematic maps on demand. In the following, we describe some of the existing work on the automation of schematic maps.

4.2.1 C-oriented line simplification

Neyer in [Neyer, 1999] describes a line simplification algorithm where the final polygonal chains must have all line segments parallel to an orientation in a given set of fixed orientations $C$ in $\mathbb{R}^2$. Typically, the number of $C$-orientations to be used is four: horizontal, vertical, and both diagonals. The simplification algorithm minimizes the number of line segments to represent the final path, which should also
stay close enough (at most a given constant distance) to the original path. Requiring the lines to be $C$-oriented helps to make the layout look graphically clearer, more structured and easier to read. Therefore, the simplification is suggested to be used in the generation of maps which are often drawn "$C$-oriented", like subway maps.

The work restricts attention to theoretical aspects of the described line simplification algorithm. In order to be useful in practice, an implementation of the algorithm with some required improvements would be necessary. Besides the fixed orientations, no other characteristics of schematic maps are considered. Topological conflicts could also result from the displacement of line segments in this simplification process, what is not considered. We also remark that in schematic maps it is not necessary to preserve equally distances of all line segments in output paths to the original locations in input paths, such as required in this algorithm.

![Original subway lines (dashed) and $C$-oriented simplified lines. At the right side, the buffer around each subway line is shaded [Neyer, 1999].](image)

**4.2.2 Discrete curve evolution**

Barkowsky, Latecki and Richter in [Barkowsky et al., 2000] present an algorithm for simplifying geographic shapes based on the method of discrete curve evolution. The main idea of this method is a stepwise elimination of kinks that are least relevant to the shape of a polygonal curve. The larger both the relative lengths and the turn angle of a kink, the greater is its contribution to the shape of a curve.

The shape relevance is measured by a cost function $k$, which is monotone with respect to the turn angle and to the lengths of the neighboring line segments. The minimum of the cost function $k$ determines the pair of line segments that is going to be substituted by a single line segment joining their endpoints. For geographic objects represented by single isolated points, e.g., cities, and for the endpoints of linear objects, no relevance measure can be computed, as there is no or only one adjacent line segment. Such point-like entities are treated as fixed points, i.e., they must neither be deleted nor moved. The points of the linear entities which can be
eliminated during the evolution process are considered as removable. When the points can not be removed but their position can be changed, they are considered movable.

This approach is performed in a context-sensitive way, i.e., before a simplification is performed, it is checked whether local ordering relations between point-like and linear features are violated. However, the final graph is still only a simplified map, just like in [Neyer, 1999]. Other constraints of schematic maps, such as fixed directions and minimum distance among objects, should also be considered.

![Figure 4.2: (a) Original transportation network and (b) the simplified network using the discrete curve evolution approach [Barkowsky et al., 2000].](image)

### 4.2.3 Grid fitting

Elroi in [Elroi, 1991] and [Elroi, 1988b] describes characteristics of schematic maps and the advantages of using GIS applications to automate the generation of such maps. In [Elroi, 1988a], Elroi presents a prototype enhancement package developed in ArcInfo with the intention of examining the possibility of automating the schematization process using GIS. He uses a method for the automation which we call grid fitting. First, the original lines are simplified, then placed on a regular grid to assure that paths are horizontal, vertical, or at 45 degrees diagonal orientations. The resulting final schemes can be later manipulated in an interactive editing session, to assure conformity to the grid, and add stations and annotation. Crowded areas can also be manually selected in the editing session, in order to be enlarged to spread density of the network.

Elroi states that the hardest aspect, during the graphic manipulation of lines, is the maintenance of the topological structure of the line network. He believes that whereas CAD systems can accelerate the graphic manipulation and aesthetic aspects of the schematization process, only a GIS can truly automate schematiza-
tion, due to its inherent topological structure. Elroi comments that his prototype has the ability to check topological integrity during schematization, but he gives no technicalities about his solution.

Figure 4.3: The grid approach to schematize lines.

4.2.4 Top-bottom order

The approach by Cabello, de Berg, van Dijk, van Kreveld, and Strijk in [Cabello et al., 2001] to the schematization of road networks is based on using a bounded number of edges per path, each edge of a given orientation (horizontal, vertical, or diagonal). They consider as input an embedded planar graph consisting of polygonal paths between specified points called endpoints. The output consists of another embedded planar graph where all endpoints have the same position as in the input graph, and every path is displayed as a two-link or three-link path, where links are restricted to certain orientations.

Globally, their algorithm works as follows. First, they replace all paths of the input map by paths of a simpler type, without creating intersections or changing topology. Then they use this intermediate map to establish a placement order for the schematized paths. The above-below relations among paths give the placement order. The order is such that they can place the schematized connections from top to bottom, each one at the topmost position that is still possible. The type of schematized paths allowed and the separation distance between two paths that do not have any shared endpoint can be specified. However, if the intermediate, simpler map does not exist, or there is no order (a cycle is detected), or the placement of some schematized path fails, the algorithm fails and no schematization is found.

The algorithm guarantees that only the four given orientations are used and topological equivalence is preserved, but a disadvantage of the method is that it does not give an output if the algorithmic conditions cannot be met. We also remark that it does not seem natural and necessary for schematic maps to have all paths in the map with a same number of links between each two endpoints. We recall that at least a general sense of the original geometry should be maintained. The recog-
nizability of similarities in a map leads the users to comprehension [Meng, 2002]. Links of length 0 could be used, but the caricature of the network would still not be based on the natural directions of the input network. We consider this a disadvantage of the approach.

![Figure 4.4: Example of schematized maps with the top-bottom approach][1]

4.2.5 Other related work

We can find computational problems similar to the schematization of lines in other research areas, for example, in VLSI layout design and graph drawing.

In the problem of arranging electronic parts for VLSI design, a graph is given where the nodes represent certain electronic objects and the edges are connections between the objects. The goal is to find a solution using minimal total area in which the objects do not overlap [Gerez, 1999].

Graph drawing is an area of computer science which deals with the development, analysis, implementation and evaluation of new algorithms for aesthetically nice drawings of graphs. Various algorithms are known. Some algorithms can bound the area of the drawing, others minimize the number of bends, or the number of edges, etc. [Tamassia and Tollis, 1995, Junger and Mutzel, 2001].

There are still other layout problems in cartography. For example, in map labeling names of places have to be placed onto a map such that no two texts overlap [Ware and Jones, 1998]. A solution is desired where the labels are closest to the points which they describe. Another application is map generalization, as given by Agrawala and Stolte in [Agrawala and Stolte, 2001]. Their work describes how to produce sketchy maps automatically, similar to hand-designed route maps, to improve the usability of conventional route maps, which are more precise and contain more information. As in our work, they also frame the layout problem as a search for an optimal layout. They use simulated annealing to search for the final layout, which should also fulfill a set of constraints, preserve topological relations and overall shape of the route network.
Chapter 5

Generating Schematic Maps

The subject of this chapter is the creation of schematic maps from a vector-based, cartographic information of a route network. An algorithm is proposed to modify positions of lines in the original input network with the goal of producing as output a schematic network that meets some common sense geometric and aesthetic constraints of schematic maps. For our map constraints we choose to create straight lines which can be horizontal, vertical, or both diagonals, and also a maximum length for a line segment and a minimum distance between pairs of map features, for readability and aesthetic considerations.

As a first step in the modification process of the original routes, geometric line details are removed via simplification. Then, the new, straightened line network is adjusted to the chosen constraints to design the schematic map. We approach this line displacement using an iterative technique driven by the map constraints. The method proposed computes a transformation which preserves topological relations among linear features using simple geometric operations and tests.

It has long been known that it can happen that iterative solutions for different problems can improve the situation initially but deteriorate after some point. This highlights the importance of knowing when it is time to stop the iterative process. Therefore, we also examine the iterative technique and provide experimental results to judge the constraint satisfaction process, emphasizing criteria related to the map quality to stop the process.

The chapter is organized as follows. In Section 5.1, we describe the process to generate schematic maps. Section 5.2 explains how to detect and avoid topological changes in the structure of the road network during the schematization process. In Section 5.3, we show our algorithm to generate topologically correct schematic maps. Then, Sections 5.4 and 5.5 describe the experiments to choose the stopping criteria for the used iterative technique. Finally, we give conclusions in Section 5.6.
5.1 General description

The purposeful deformation of a geographic map is a common operation in cartography. In general, two types of map deformations can be considered: one for bringing two maps together, such as combining a satellite image and a geographic map via rubber-sheeting or minor geometric adjustment, and the other, to highlight quantities other than geographic distance and area, such as in cartograms [Dey et al., 1998]. Our schematic maps can be thought of as the first type of deformation, but we use techniques for transforming lines similar to [House and Kocmoud, 1998] and [Dorling, 1996] for constructing continuous area cartograms.

To begin the automatic generation of schematic maps, we assume a database with the line network to be schematized is provided as input to our algorithm. We choose for our maps the following constraints: straight lines at angles of 45 or 90 degrees, a user-defined maximum length for line segments and a user-defined minimum distance between pairs of segments.

As a preprocessing step, we perform point and line filtering to reduce the number of visualization points that constitute the detailed route network. We use the Douglas–Peucker algorithm to remove line crenulations [Douglas and Peucker, 1973]. This simplification algorithm is well-known and flexible, with a user definable threshold that controls the amount of simplification to be performed. After this simplification, we cut too long straight lines using the maximum length constraint.

The next step is to find a new location for each point of the simplified lines according to the other constraints of the schematic map. The network conformation is improved iteratively, in order to satisfy the map constraints (here related to angles of lines and distance among segments). We used an iterative algorithm based on a gradient descent technique [Müller, 1999] driven by the map constraints [House and Kocmoud, 1998]. The algorithm visits all points \( p_1...p_n \) of the simplified map in turn to change the actual conformation. If the current point \( p \) violates one or more constraints, a better location \( p' \) for \( p \) is computed. Without any stopping criterion, the user decides when the schematic network reaches an acceptable appearance to stop the process.

For the constraint of stylized angles, say \( c^\phi \), the new location \( p^{(j)} \) is chosen among the locations that fit \( p \) in one of the allowed schematic directions. Note that in each line segment with \( p, \) e.g., \( \overline{pq} \), we use the middle point of the segment to compute the distance to the nearest schematic direction. This is done, in order to have smaller, finer displacement vectors than we would have considering the distances from the
other endpoint of the line segment [Müller, 1999]. The nearest distance \( \Delta p \) determines the new location for \( p \) (see \( \overline{pq} \) in Figure 5.1a).

For each constraint \( c^i \in \{c^1, \ldots, c^m\} \) that affects \( p, p^{(i)} \) is the location nearest to \( p \) that satisfies \( c^i \). The constraints evaluation is carried out for all line segments in which \( p \) exists. The new location \( p^f \) for \( p \) will be obtained by the arithmetic mean of all required locations \( p^{(i)} \) for all line segments with \( p \) (in Figure 5.1b, three constraints are considered). Although, instead of computing a simple mean of the displacement vectors, if certain roads are more important than others, it would also be feasible to evaluate a weighted average of the vectors.

\[
\begin{align*}
\Delta p & = p^{(j)} - p \\
\frac{1}{n} & = \frac{1}{n} \sum_{i=1}^{n} \Delta p_i
\end{align*}
\]

Figure 5.1: (a) Fitting a line segment \( \overline{pq} \) to the nearest schematic direction and (b) finding the new location \( p^f \) for \( p \).

The minimum distance constraint is the last to be checked. The final location \( p^f \) should keep the specified minimum distance to the other linear features of the map. Adjustments in the arithmetic mean of the final displacement \( \Delta p^f \) vector are then performed when necessary. Line segments are stretched or shortened to reach the new computed locations, which we refer to as schematic locations. However, the topological structure of the line network has to be preserved during this modification process.

\section*{5.2 Preserving map topology}

The previous section described how to compute a better location \( p^f \) (better according to the aesthetic and geometric constraints) for a given point \( p \). By moving a point in a line network, it can happen that we reach situations in which the map topology changes. We propose a context-sensitive displacement, i.e., before displacing a point from its original position \( p \) to a new location \( p^f \), a test is performed to detect the context situations that can lead to topological errors in the line net-
work. In this section, we describe such a test and also how to adjust, if necessary, the new location $p'$ to avoid change in the map topology. Map topology here means the following properties:

(P1) no absence of line crossings that are present in the input map;

(P2) no line crossings that are not present in the input map;

(P3) cyclic order of outgoing connections around any node agrees with the ordering of connections in the input map.

Analysing the displacement of a point $p$ in a line network, we can observe that a triangle is formed with $p$, the new location $p'$, and the other endpoint $q$ of the line segment with $p$. If a topological change occurs when moving $p$, this will happen inside the triangle $T(pqpq)$. Thus, we need to find out what situations can happen inside this triangle. Figure 5.2 illustrates some situations. Basically, the triangle $T(pqpq)$ can contain one or more line segments crossing any of its sides (Figure 5.2a), or it can contain inside it one or more points with different degrees (Figure 5.2b-f). Cases which can lead to a change in the map topology were identified. We use these cases to perform a test to preserve the map topology. We explain the test first and prove its correctness later.

The test to be performed in order to preserve the map topology will involve all line segments which have $p$ as an endpoint. It is sufficient to consider a single such segment, say $pq$, to describe the test. Consider the triangle $T = T(pqpq)$ to perform the test. We have to find out whether there is any line segment of the map crossed by the boundary edge $pqp$ of the triangle $T$ (see [de Berg et al., 1997] for algorithms to find line intersection). We also have to check whether the triangle $T$ contains inside it any point which can lead to a change in the map topology [Avelar, 1997].

If the test finds that topology will change, the move of $p$ must be smaller than the whole displacement to $p'$ to avoid the change. It is important whether a point moves at all, because the displacement can affect the adjacent line segments of the point in analysis, such that the modification process of the network keeps trying to find a better location for all points.

We distinguish between the following cases in our topology test:

(1) There is no point inside the triangle $T$ and no line segment crossing edge $pqp$. 
In this case, map topology will not be changed. The move of $p$ to $p'$ is allowed. See Figure 5.3.

(2) There is at least one line segment intersecting edge $pqp$. 
In this case, map topology will change, as illustrated in Figures 5.4a and 5.4b.
Figure 5.2: Some possible situations when displacing a point. (a) The formed triangle \( T(pp'tq) \) does not contain any point inside it, but a line segment crossing two of its sides. (b)-(f) The triangle \( T(pp'tq) \) contains one point inside it, which, as a node, is presented having, respectively, degrees 1, 2, 3, 4 and 5.

Figure 5.3: Point \( p \) can be moved to \( pt \).
A new location for $p'$ has to be obtained. The new $p'$ is the nearest intersection point, say $u$, to $p$ plus the minimum distance constraint $d$ measured along $pp'$ (or minus $d$, according to the direction of the displacement). See Figure 5.5a.

Figure 5.4: Possible cases for which topology changes, when displacing $p$ to $p'$. In (a) there is a line segment crossing $pq$ before the displacement takes place, but not any more $p'q$ after it. (b) There is no line segment crossing $pq$ before the displacement, but $p'q$ after it. In (c) and (d) the point inside the triangle is endpoint of one or more segments which also have crossings changed after the displacement.

(3) There is at least one point $v$ inside the triangle $T$.

In this case, map topology might change. See Figures 5.4c and 5.4d. To calculate the new location for $p'$, we define a straight line $l$ through $v$ and $q$, and calculate the intersection point $u$ of $l$ and the edge $pp'$ of $T$. Take $p'$ to be the nearest intersection point $u$ to $p$ plus (or minus) the minimum distance constraint $d$ (see Figure 5.5b).

Note that, after the above checks have been performed, the newly adjusted location $p'$ will be such that case (1) holds for all line segments $pq$. In other words, the situation in Figure 5.3 holds.

The cases (2) and (3), as well as more cases of each one, can occur together. In such situations, the new location for $p'$ will also be computed taking the nearest intersection point to $p$. 
5.2 Preserving map topology

Lemma 1 If case (1) above holds for all line segments \( \overline{pq} \), then moving the point with location \( p \) to the new location \( pt \) will preserve properties (P1), (P2) and (P3) described in Section 5.2.

Proof. We consider each property in turn.

(i) Since only point \( p \) is moved, we only need to consider crossings with line segments of the form \( \overline{pq} \). Assume there is a line segment \( \overline{ab} \) which crosses segment \( \overline{pq} \). Because, by assumption, neither point \( a \) nor \( b \) lies inside triangle \( T(pptq) \), and \( \overline{ab} \) does not cross \( \overline{ppl} \) since case (1) holds, segment \( \overline{ab} \) must also cross \( \overline{ptq} \). See Figure 5.6.

(ii) Note that triangle \( T(pptq) \) coincides with triangle \( T(ptpq) \) and line segment \( \overline{ppl} \) coincides with segment \( \overline{ptp} \). The result now follows from property (i) and from symmetry by considering the transformation of the point with new location \( pt \) back to the old location \( p \).

(iii) See \( p \) and \( q_i \) in Figure 5.3. For nodes like \( p \) a displacement of \( p \) to \( pt \) could change the cyclic order around \( pt \), but because no point, in particular, \( q_1, \ldots, q_i \), can be inside the triangles \( T(pptq_i) \), a change in the circular order is impossible.

Now consider nodes like \( q \) in Figure 5.7. Suppose that \( \{\overline{qp}, \overline{qp_1}, \overline{qp_2}, \ldots, \overline{qp_k}\} \) is the set, in cyclic order, of all outgoing connections from node \( q \). Provided that angle \( \gamma \) is less than \( \theta \), the cyclic order will be preserved; but this must hold since otherwise case (1) would not be satisfied.
Generating Schematic Maps

We remark that although case (1) is sufficient to preserve map topology, it is not always necessary. Consider Figure 5.8. Since there are no line crossings in $pq$, nor in $p'tq$, the topology is preserved. However, we consider using the strong case (1) a positive feature of the triangle test, since we want, whenever possible, to avoid changes in the "sidedness" relations among geographic elements, as represented by $pq$ and $v$. The test of point inside triangle can also detect some of such sidedness changes among geographic features, as shown in the Figure 5.8 ($p$ is initially at one side of the lake and $p'$ is at the other side). Extensions of this solution could also check the neighbourhood along $ppl$ outside the triangle.

Figure 5.8: Topology is preserved, but not "sidedness" relations.

5.3 The algorithm

We next present the algorithm to generate topologically correct schematic maps.
5.3 The algorithm

Procedure SchematicMap (map, maxSegment, tolerance);
Input: N points of the conventional map, the maximum length of a line segment,
the simplification tolerance
Output: n points for the schematic map
begin
simplify lines using Douglas–Peucker algorithm with user-defined tolerance;
cut long line segments according to maxSegment;
repeat
for all points $p_1...p_n$ of the map do
 test constraints and calculate a new schematic location $p_i$ for $p_i$;
 if there is any segment collision with $p_i p_j$ then
 calculate new location for $p_i$;
 endif
 for each point $q$ that makes a line segment with $p_i$ do
 if there is any point inside triangle $p_i p_j q$ then
 calculate new location for $p_i$;
 endif
 end for
 change location of $p_i$ to $p_i$;
end for
until reached desired appearance;
end.

As previously explained, the search starts with the original configuration of the route network. The network is iteratively improved by performing optimization steps based on different moves in the inner loop. We do not need to test all line segments and points in the map to look for geometric intersections of line segments and points inside a triangle, because the nodes and segments are stored in a uniform grid similar to the data structure described in Section 3.5.3. A grid divides the plane into non-overlapping regions. Line segments are distributed among the regions where their center point lies. The grid spacing is here chosen such that the line segments are at maximum the size of a cell. In this way, potentially intersecting segments can only be found in either the same cell or two neighbour cells. Therefore, only adjacent regions of the point being analysed need to be considered when searching the data. Spatial partitioning algorithms like this are sufficient, because a transport route network is spatially well distributed, and we do not move points too far [Andrews et al., 1994].
5.4 Stopping criteria for iterative assignment

We described previously how schematic maps are generated using an iterative improvement technique based on the simple displacement of map objects. The algorithm tries to schematize the line network by evaluating a number of local moves for each point and combining them to give a single displacement for the point. Empirically, the iterative process yields good results in the visualization of schematic maps, but when doing iterations we need to find a stopping criterion. The iterative process could continue until no further improving move can be made or until it meets some user defined stopping criterion. We have seen that it cannot be guaranteed that an optimal solution will be found. Realistically, the search is for an acceptable, rather than the 'best' map for a particular dataset. In this section, we provide an investigation that leads to the issue of stopping criteria to the chosen iterative assignment for the generation of schematic maps. This requires a convergence evaluation of the iterative process, which will also be presented.

It is intuitively clear that iterative algorithms have a beneficial effect on the structure of the schematic routes. It is less clear how to describe exactly where such iterations lead: Do they converge? If so, do they converge towards a fixed point? Or towards something else? When is a resulting map better than another one? Is this a viable alternative for generating schematic maps at all? What are the stopping criteria?

In this investigation, we aim at answering these questions. A key requirement in all heuristic search applications is a willingness to experiment. Each iterative algorithm uses a specific termination criterion, depending on the application.

We know that there is a huge number of possible ways of schematizing routes even for a small database. Therefore, we do not have a single map solution previously known. Given the numerous ways in which a schematic network may be represented, some criteria are then needed for determining whether a map can be regarded as acceptable for its intended use. The measure of acceptability of each candidate map determines the quality of that map state. We can formulate the criteria determining map quality as an evaluation function and use it to compare different possible maps of a given dataset [Lonergan and Jones, 2001]. The set of all possible states in which a schematic map can be, considering each individual object state (including the ones resulting of different simplifications), constitutes the search space.

To choose the stopping criteria for a certain application is a matter of empirical experimentation. For example, we can apply the iterative method in a situation where the correct result is known beforehand. In such a determinis-
tic steady-state assignment\(^1\), the system is just monitored to stop when changes (according to a pre-defined measure) are smaller than some pre-defined solution [Kelly and Nagel, 1998]. Another conventional stopping criterion is given by a function of the number of iterations. We can iterate the algorithm a number of times and observe the quality of results, finding out when a result can be considered acceptable.

In order to choose the stopping criteria for the iterative algorithm used, we need to find out in our experiments which constraints can characterize when a map is acceptable. We need also to formulate these constraints in an evaluation function. Ideally, we can formalize when a resulting map is better than another one, i.e., given two maps that convey the same information, we look for the definition of a function \( f: \text{set of maps} \rightarrow \mathbb{R} \), such that \( f(\text{map}_1) > f(\text{map}_2) \), if \( \text{map}_1 \) is better than \( \text{map}_2 \).

### 5.4.1 Analysing map quality

Lonergan and Jones in [Lonergan and Jones, 2001] point out that the way a measure of map quality is best defined will depend on particular circumstances. The simplest definition will produce a Boolean result; either the map is good enough or it isn’t. While this definition is sufficient for deciding whether a single map is acceptable, it gives no assistance in finding such a map. For this, information on how and where the map is inadequate is necessary.

A central difficulty of optimization in cartography is that the 'visual appearance' of a map depends on the interrelationships between all the data points in the map. For the purposes of reducing the enormous number of possible maps, it may be appropriate to identify where and between what features visual clashes are likely to occur.

We generated schematic maps of a test dataset for a different number of iterations, in order to better understand what the iterative algorithm is doing. With this experiment, we aimed to identify in practice tendencies and undesirable characteristics in the resulting maps. We recall what we want in our schematic maps: the topology has to be preserved, while directions and distances can be approximately preserved. Figures 5.10 through 5.14 show resulting maps for 10, 100, 1000, 10000 and 100000 iterations.

---

\(^1\)In a deterministic model, all mathematical and logical relationships between the elements are fixed. As a consequence, these relationships completely determine the solutions. We say that a system is in steady state, or in equilibrium, if the probability of being in some state does not vary in time.
Figure 5.9: Original input network.

Figure 5.10: After 10 iterations.

Figure 5.11: After 100 iterations.

Figure 5.12: After 1000 iterations.
Figures 5.13 and 5.14: After 10000 and 100000 iterations, respectively.

Analysing the resulting maps, we can see that the best result is not obtained with the largest number of iterations. Actually, iterating a large number of times seems like a waste of computational resources, since there is no reason to believe that good maps go to a state like the resulting map presented in Figure 5.14. The reason is that the algorithm optimizes for angles, but does not look at preservation of distances. The angles in Figure 5.14 will be slightly better than in Figure 5.10, but the relative distances (distortion) are a lot worse. In order to make a list of situations which should not occur in the final schematic map, we have identified problems in these resulting maps. Figure 5.15 illustrates such problems. We can also find some of such undesirable characteristics in hand-drawn schematic maps, e.g., in Figure 2.13. We now give some more information on the listed characteristics:

1. We can observe that for a large number of iterations, e.g., 10000 or 100000, the length of line segments becomes generally much longer or much shorter than in the original network. Such large deformations should be avoided, since, for recognition purposes, line segments should keep lengths approximately similar to their original values.

2. Adjacent line segments should not be displaced to the same position after a transformation, especially when line segments are part of curves and bends. For a more adaptive and realistic cartographic design, curves should be treated individually. The user has to decide how is better to change the

**Figure 5.13:** After 10000 iterations. **Figure 5.14:** After 100000 iterations.
Generating Schematic Maps

Figure 5.15: Characteristics to be avoided in the resulting schematic maps.

<table>
<thead>
<tr>
<th>Input map → Schematic map</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td>Too long line segments.</td>
</tr>
<tr>
<td><img src="image2.png" alt="Image" /></td>
<td>Adjacent segments displaced to the same position.</td>
</tr>
<tr>
<td><img src="image3.png" alt="Image" /></td>
<td>Changes in spatial relations among geographic elements.</td>
</tr>
<tr>
<td><img src="image4.png" alt="Image" /></td>
<td>Line segmentation, loosing continuation of lines.</td>
</tr>
<tr>
<td><img src="image5.png" alt="Image" /></td>
<td>Minimum distance should not generate parallel segments.</td>
</tr>
<tr>
<td><img src="image6.png" alt="Image" /></td>
<td>Too small angles.</td>
</tr>
</tbody>
</table>
original line segments of a curve, according to the neighborhood of the curve. For example, for one of the cases illustrated in Figure 5.16.

![Figure 5.16: Some possible designs for schematizing a curve.](image)

3. For recognition purposes, spatial relationships like above-below and left-right relations among geographic elements should also be kept. The proposed algorithm for preserving map topology can identify only some of such spatial relations involving sidedness of geographical elements. This topic has been only recently studied in cartographic generalization, for example, in [Hardy, 2001].

4. The problem of loosing the natural continuation of a line in a road network is also a recent subject of study. Some progress has been made on it in cartographic generalization, e.g., in [Thomson and Richardson, 1999].

5. As mentioned in Section 4.1, features must be separated from each other by at least a given minimum distance in the final map.

6. The line network should not contain too small angles, keeping the computed schematic directions as much as possible. This is similar to item 2, but not the same, because objects are not displaced to one same position.

Such undesirable characteristics in the final schematic maps may determine new constraints, which can have a clear intuitive semantic to the map designer. Each of the new constraints can be considered in the choice of an appropriate map quality measurement. A useful measure of quality, or of acceptability of a map, needs to be unambiguous, objective and repeatable. Ideally, the measure of quality should be good at discriminating between potential solutions and quick to calculate [Lonergan and Jones, 2001].

On a general level, we consider a map with too many undesirable situations unacceptable. The better of two intermediate maps is obviously the one with fewer undesirable situations. In terms of constraints, an acceptable solution should agree, as
Generating Schematic Maps

much as possible, with certain design characteristics. We know that any map which completely preserves map topology, and approximately directions and distances, is acceptable. To keep approximate directions, a map should have line segments with minimum error in the direction of line segments in comparison to the calculated schematic direction for the segments. We face then a minimization problem of a function of residual error. The other constraint for the quality measurement is the length of line segments, which should be approximately similar to the original network. The map quality in this work will then refer to the constraints of angular error and of length.

We considered more informative to work with these quality constraints separately, in spite of using representations of them in equations. In the following, we provide the analysis of convergence of the algorithm for the referred map quality constraints.

5.4.2 Convergence evaluation

We need to study the convergence of our iterative algorithm, because the iterative approach that we used is neither steady-state nor deterministic, and it is unclear if any of the them holds.

For cases in which convergence assumes strong convexity, i.e., the optimal solution is unique, the analysis of convergence becomes relatively easy. Since there is a unique solution satisfying the optimality condition, the convergence of the algorithm follows [Openshaw and Openshaw, 1997]. This is not applicable to problems without strong convexity or when the objective function is nonconvex.

Realistic nonlinear functions tend to be complex, perhaps discontinuous, and generally not well behaved, which often means that standard mathematical optimization methods can probably no longer work, optimality can no longer be proven, and various numerical instability problems can be caused [Openshaw and Openshaw, 1997]. However, in spite of this lack of assurances of optimality, a result might be good enough without being the best one globally, as explained in Section 4.1.2. It turns out that error bounds can help to establish convergence in such cases [Luo, 2000].

To evaluate the convergence of our iterative approach, we select some constraints and relax them. Relaxing refers to making small adjustments to the position of objects such that the constraints are satisfied. A relaxed system should somehow reflect a different balance between the constraints than an unrelaxed system or the original network. Our intuition is that in a nice map, “in equilibrium”, routes would be distributed more evenly among space than in the results shown in
Figure 5.13 and 5.14, and angles will be nearer to schematic directions than in the original network. However, in general, it is difficult to find out which of the constraints are the best to choose for relaxation. As explained in the previous section, we chose here the constraints interesting to analyse the map quality.

The evaluation of an objective function is carried out using separately: the angular error between the final direction of a line segment and the calculated schematic direction for it, and the line segment length. Additionally, to follow the amount of displacement according to the progress of the iteration, we also use the final displacement of an object.

Focusing on the same study area of Figure 5.9, we studied the selected constraints for 1000 iterations. Using graphs, we plot the sum of all angular errors of line segments per iteration (Figure 5.17), and the sum of all lengths of line segments versus the iteration number (Figure 5.18). We also plot the sum of the amount of displacement of all objects moved in each iteration (Figure 5.19).

![Figure 5.17: Sum of the angular error of all objects.](image)

We can observe in the graphs of Figures 5.17 (angular error), and 5.19 (displacement) that the schematization improves rapidly during early iterations followed by a degradation in large number of iterations. The zero iteration network is far from relaxation, and the 100th iteration is more relaxed, although not completely.

At the beginning of the iteration process, the original location of routes makes the sum of all angular errors and of all displacement vectors large. With the increasing number of iterations, we can see that the total angular error and the total displacement of the dataset decreases. With the total displacement, we can just say
Figure 5.18: Sum of the length of all line segments.

Figure 5.19: Sum of the displacement vector of all objects.
that the amount of changes on it also goes down. This is not necessarily good to prove that the map quality improves. But the decrease in the angular error means that the map quality improves.

Note that, in Figure 5.17, the angular error after the first 100 iterations is not much higher than what is reached near the end. Extensive experiments indicated that this measure may not converge to zero, indicating that there are indeed considerable variations between iterations, which just cancel out in the aggregate variable used. Thus, at best one could demand that this value converges in the average. After 400 iterations, the angular error appears to be measuring a network's proximity to equilibrium.

Contrarily, the total length of segments increases with the number of iterations (Figure 5.18). The total length error indicates that the overall map quality does not only improve for a large number of iterations. We can see this in Figure 5.20, where we show a graph with the total length error of the dataset measured for a larger number of iterations.

Considering the graph of total displacements in Figure 5.19, note that the curve is not yet completely flat after 100 iterations, indicating that the system is not yet completely "relaxed" in respect to this constraint. Nevertheless, the value of the error function was changing very slowly after this point such that further iterations resulted in little change in the schematized map.

With this convergence evaluation of the constraints chosen to measure the map quality, we decided to use the angular error in the stop criterion of the iterative process. The reason is that we mainly want to produce maps in schematic directions and this parameter is a main required characteristic for the final map.

### 5.4.3 Stopping criteria

By experimentation, we have found that 200 iterations can provide an acceptable result for our schematic maps. This particular value is also justified by results from other iteration series which indeed seemed to have no clear improvement any more. Thus, we set the maximum number of iterations to be 200 as a simple stopping criterion for our algorithm.

We also chose another termination criterion for the iterative approach based on the map dataset itself. The criterion is based on the angular error constraint. Considering that in a schematic map the information content of long lines can attract more attention than of small segments, it is reasonable that the angular error of long
Figure 5.20: Sum of the length of all line segments for the same data, but with the number of iterations increased by 4000.

Line segments have more impact on a stopping criterion.

To have a measure that is consistent across different datasets, the measurement of the angular error should be in some way normalized. In this case, the angular error $a$ of each line segment is divided by the maximum angular error value, that can not be exceeded on any map state ($\text{max} = 22.5$ degrees). This gives the normalized angular value $e = a/\text{max}$.

We can sum the products of the length $l$ of each straight line segment and its normalized angular error $e$ and divide this summation of $(l \cdot e)$ by the total length $\text{totl}$ of all line segments on the map. The total angular error is then

$$\frac{\sum(l \cdot e)}{\text{totl}}.$$

If all line segments have maximum angular error, i.e., $e = 1$, the total angular error would then be equal one. For the maximum tolerable error, we choose a value between 0 and 1. We considered enough to stop the process when the angular error is 10% of the total error value, that is, for reaching an acceptable result in this thesis ninety percent of directions should be schematized:

$$\frac{\sum(l \cdot e)}{\text{totl}} < 10^{-1}.$$

Note that this approach is not well-justified from a behavioral perspective, because it would imply that mapmakers optimize their behavior so that the expected angular error would be always minimized to 10%. In other situations, it could be
that forcing the algorithm to generate averaged behavior does not necessarily lead to good results [Kelly and Nagel, 1998]. We also call attention to the fact that the iterative algorithm only minimizes with respect to the last iteration. Cartography rules can provide more information in the future to make the iterations and results more realistic.

In general, the algorithm reached 200 iterations before the total angular error was equal to 10%. The value of the total angular error function was changing very slowly at this point, so that further iterations resulted in little change in the resulting schematic map.

## 5.5 Conclusions

The main ideas of the schematic map generator were implemented with encouraging results. For this study, map labels and cartographic features were not included in the visualization.

For the line simplification, we used the classic Douglas-Peucker algorithm. However, we call attention for the fact that there could be some topological and self-intersection errors introduced by using this algorithm. Variants of the Douglas-Peucker algorithm which take topology into account [Saalfeld, 1999] or other line simplification methods [de Berg et al., 1995] could be tried instead.

We described an iterative strategy for the displacement of points in a network. We also presented a measure of map quality that can be used in the search for an acceptable schematic network of a given route database. We have also developed a new method for preserving topological relations among linear features while generating schematic maps. The method can also be applied in other situations, for example, when schematic maps are generated using the grid fitting method [Elroi, 1991].

The iterative method is driven by common sense manual displacement rules that are expressed by constraints. Because the constraints can be contradictory, the iterative method uses their mean to find a compromise solution. The optimization technique was the gradient descent. The method has shown promising results in applications with a few object types. However, a typical demanding aspect of search procedures is the need to evaluate numerous alternative states of the application. This requires efficient procedures to perform tasks related to the selected set of constraints of the map, such as finding and measuring distances to nearest neighbours of a given map object and treating overlap between map objects.

The iterative method does not attempt to produce the best possible map of a situation, but merely an acceptable one. Ideally, acceptable schematic maps should:
• reflect route directions adequately,

• respect constraints,

• be comparable in quality to previously published schematic maps.

All the decisions about the stopping criteria are important and far from straightforward. Effectively handling real-world constrained problems is a challenging task [Michalewicz and Fogel, 1998].

We have also shown examples of schematic results obtained with the iterative approach. Results can easily be improved further by post-processing. Users can eliminate undesirable characteristics, include additional information, or modify lines, in combination with other individual priorities for the design of the map. This can be done using any graphic software.

Finally, the iterative method can also be extended with new constraints if the ones that are used in this thesis are not sufficient for a particular displacement process. For example, considering the minimal dimension of a linear object.
Chapter 6

Experimental Results

In the previous chapters, we have introduced the design, modeling, and generation of schematic maps on demand. In this chapter, we present experimental results to show the applicability of the work. We schematize automatically the route network of a selected area in the city of Zurich and produce a cartographic schematic map. We also describe the experimental design developed for a project with the purpose of exploring new possibilities of schematically showing the regional train and bus network of the canton of Zurich.

First, we describe our experiment of producing a schematic map using a transport database. We begin by presenting the data used and the geographical context in Section 6.1. In the same section, we give results, evaluate the time to produce them and comment the choice of map design. Then, we present the design project of a new schematic map for the public transport network of the canton of Zurich in Section 6.2. Finally, we provide discussions in Section 6.3.

6.1 The experiment

In this experiment, we carry out a case study with real world data. We focus on a study area within Zurich which has a perimeter of approximately 2500 meters by 3000 meters. The focused area is located around the busy intersections of the Bellevue tram station and the main station, both near the lake of Zurich. We produced a cartographic schematic map for the study area using the information and approach in this thesis. We chose this area of Zurich because it presents an intricate circulation system and it has an adequate official schematic map. Figure 6.1 shows the study area represented in a city map [Orell, 1998]. The same area is shown in the official schematic map of the public transport authority of Zurich [VBZ, 2001] in Figure 6.2.
Figure 6.1: The study area in a city map of Zurich. Source: 'Mobil in Zürich', design by Orell Füssli Kartographie AG Zürich [Orell, 1998].
Figure 6.2: Our study area in the official schematic map for transportation of the city of Zurich. Source: 'VBZ Züri-Linie', design by A+H Eggmann SGV AGI [VBZ, 2001].
6.1.1 The data

The data considered for this experiment include all streets with car traffic in the selected study area. The street network includes the coordinates of the start and end points of each street, control-points related to their shape, and intersections. The data for the street network were offered for academic use by the company Tele Atlas AG. The whole dataset of Zurich contains 59904 roads, 62034 nodes and 123496 shape points. The data of our study area has 609 roads, 1102 nodes and 238 points. Figure 6.3 shows the street network. Main streets are shown in red, secondary streets in black, and side streets in blue.

Figure 6.3: The street network of the study area to produce a schematic map.

The data format of the transport data set is the GDF (Geographic Data File), which is a relatively new spatial data format standard adopted in 1995 by the European Committee for Standardization (CEN). This format has been developed in a
European project named European Digital Road Map (EDRM), in order to be used to describe and transfer road networks and road related data. GDF provides rules for capturing road data and defining features, attributes and relations [CEN, 1995]. The primary use of GDF is for car navigation systems (e.g., Bosch, Philips, Volvo, etc.), but it can also be used for other specific transport and traffic applications with routing, location services, and GPS.

GDF is not only a theoretical standard in Europe, the format has been pushed by the major digital road data suppliers: EGT, Bosch, ETAK and Tele Atlas. These companies have committed themselves to building their databases according to GDF specifications. However, one limitation related to GDF is that at present there are no available commercial GIS conversion tools for this format.

Regarding the current situation of conversion tools of GDF, we had experience with the conversion of ArcInfo data to GDF. A database based on the data model described in Chapter 3 was created in the winter of 2000, using ArcInfo version 8.0.2 [Purtschert, 2001]. The location coordinates of some streets with transport routes, stations, part of the lake of Zurich, and some reference places on the same region of our study area were stored. The information was taken from the VECTOR25 dataset of the Swiss Federal Office of Topography, which presents vectorial cartographic information extracted from a pixel map in the scale 1:25000. The conversion to GDF was provided by the company GISCON Jung + Richter Gbk from Munich, which developed the own tool to convert data in ArcInfo to GDF. But the final results were not completely compatible to the GDF standard used by Tele Atlas. For example, in the description of route features, the street segments taking part in a route are not listed in the order in which they appear in the routes.

6.1.2 Results

The pre-processing step of the schematization process is the simplification of the line network. The simplified network of the study area is shown in Figure 6.4. Recall that the Douglas-Peucker algorithm has been used.

As we have seen in Chapter 5, by distorting road lengths and angles, there exists a large space of possible schematized routes. The system performs a search over this space to find a layout for the map. The rationale is to return only an approximate search result, and to compute it in a significantly shorter amount of time. With the stopping criterion of 200 iterations, the schematization of the data gives the result presented in Figure 6.5.
Figure 6.4: Simplified street network using the Douglas-Peucker algorithm.
Figure 6.5: Resulting schematic lines with the approach of this thesis.
We analysed the performance of our implementation for completing the 200 iterations. We used a Java 1.1.8 compiler and measured runtime on a Pentium III 1000 MHz processor. Figure 6.6 shows the running times of the iterative algorithm. The vertical axis gives the average CPU time (in seconds) and horizontal axis gives the iteration number. Each iteration takes on the average 1.5 seconds. It took about 4.5 minutes to compute the schematization of Figure 6.5. Actually, after 50 iterations, the network has already relaxed to a point where link flows and turn counts resemble those seen in schematic maps, but there were still some lines that were not schematized.

![Figure 6.6: Running time of each iteration for the study dataset.](image)

![Table 6.6: Running time of each iteration for the study dataset.](table)

Figure 6.7: Total running time for the study dataset.
The sum of the execution times of the iterations is shown graphically in Figure 6.7. This figure tells us that the runtime behavior for the data is linear in the number of iterations.

Using the information provided in Chapter 2, we also designed the resulting schematized line network. We used a drawing tool to cartographically symbolize transport information of the route network. We chose to work with Freehand 8.0, because this graphic software has been utilized as a basic tool for cartographic drawings at the Institute of Cartography of ETHZ. Extra functions were especially implemented to adapt the software for cartographic applications.

After a preliminary inspection of our schematic results, we found that the network lines have a similar design appearance and feel to a map in classic style, such as the Porto bus map presented in Figure 2.8. Figures 6.8 and 6.9 show two variants of schematic maps designed in the classic style. In the map of Figure 6.8, we chose rather a simple symbolization. We have only changed the width and color of the transport route lines and added the labels of transport services and stations at the streets they occur. In the map shown in Figure 6.9, we basically changed the width of all lines and added color to the blocks around the lines. We have additionally drawn the routes on the streets in which they are located, and also added color to the lake and to some vegetation areas in the background. After adding cartographic features and context information on this map, we hoped to achieve a result of comparable quality to the Porto map in Figure 2.8.

We can observe that our schematic maps are more similar to the original network in Figure 6.1 than the official map of Zurich for the same area illustrated in Figure 6.2. The advantage of it is that the general appearance of the network is kept, and similarities help the map user to find places [Meng, 2002]. In Figure 6.2, we can also see that certain regions with too dense information were exaggerated and present more schematization.

6.2 The ZVV project

Description of the project

The Zurich Transport Network (ZVV - Zürcher Verkehrsverbund) contacted ETHZ for the development of a project with the purpose of redesigning the current schematic map showing train and bus routes in the canton of Zurich. As in classical schematic maps, the regional Zurich map includes only the most important rivers and lakes in the background. However, because of the special topography in the canton of Zurich, ZVV would like that users also have an idea of important topo-
Figure 6.8: One possible design of a schematic map in the classic style.
Figure 6.9: Another schematic map in the classic style, but similar to the Porto bus map shown in Figure 2.8.
graphic features of the region, namely terrain information, vegetation, and towns. Thus, the new experimental design should include generalized topographic elements in the map’s background. Most importantly, the project should show if a new design can be found that could completely replace the current schematic map, or if the current schematic map should just be updated.

The project took two months to complete in the spring of 2002, and it was executed under the direction of the Institute of Cartography of ETHZ [Hurni, 2002]. In the following, we give an analysis of some experimental results and present one final schematic map proposed in the project.

The current ZVV schematic map

Figure 6.10 illustrates the current version of the regional schematic map with the train and bus network of the canton of Zurich [ZVV, 2000]. This map is similar in style to the tram map of the city of Zurich previously illustrated in Figures 6.2 and 2.7. All lines are drawn with 0, 45 or 90 degree angles. Bus lines have the classic style. They are all in the same color, and are thinner than the train lines. Train lines are drawn parallel to each other and they are all presented in a different color – French style. Train lines have circular arcs around bends, while bus lines have adjoining straight lines. Bus stops are symbolized by full circles in the same color of the bus line. Train stations are represented by white circles. When there are two or more lines in a train station, it is represented by white cylinders.

Labels for identifying transport lines and stations vary in three different sizes, being written in 0, 45 or 90 degree angles, according to the available space. Train route labels appear in special insets along or beside a train line.

A simplified shape of the canton of Zurich appears in the background, which also contains highly generalized shapes of the most important lakes and rivers of the canton. The airport is the only reference place whose location is included on the map.

Experimental design

As a test area, the region of Limmattal/Furttal, on the west side of the Zurich main station, was chosen. This area contains many transport lines, connections and stations. Figure 6.10 shows exactly this area of the existing schematic map. The area is also shown on the topographic map in Figure 6.11.

For this project, lines were not automatically schematized, since there was no appropriate transport database available for it. In order to schematize lines and redesign the map, the vectorial, cartographic data VECTOR200 from the Swiss Fed-
6.2 The ZVV project

Figure 6.10: Example of the schematic map showing the public transportation network of the canton of Zurich. 
*Source:* 'ZVV S-Bahn und Busse', design by Zürcher Verkehrsverbund [ZVV, 2000].

Figure 6.11: The test area represented on a topographic map of the canton of Zurich. 
*Source:* Verkehrskarte des Kantons Zürich, design by Orell Füssli Kartographie AG Zürich [Orell, 1990].
eral Office of Topography were used. With the information in this dataset, the schematic map can also be legible in small formats.

As explained in Chapter 2, there are many possible variations for the design of a schematic map, considering different styles, levels of content, colors, and symbolization. After some meetings and discussions with ZVV, certain aspects appeared as desirable to the new design. For example, the blue color of bus lines in the current map was chosen to be kept, and the new schematic map should be printed on paper, in the three existing sizes currently used by ZVV. The background of the new map for ZVV should still contain water features, but also include forests, villages, and some terrain information.

As described in Chapter 2, in general, schematic maps include only a stylized representation of lakes and rivers in the background. However, in Figure 2.9 we gave an example of a subway map which presents other features than only lakes and rivers on the background. Figure 2.9 shows metro lines of Madrid on a background including vegetation, buildings, and main streets. Main roads and highways could also be included, as the Copenhagen bus map illustrated in Figure 2.10. Schematic maps for transportation do not usually contain much information about the terrain.

A good tactic for determining a map design is to prepare alternatives to explore the mapping possibilities, and then select the most appropriate for the intended task. We show in Figures 6.12 and 6.13 two drafts prepared for studying the design of the ZVV map. The assignment of symbols to map features can lead to endless possibilities for colors (including hues, saturations and intensities), sizes, positioning, font types, etc. A list with some aspects that should be taken into consideration in each situation is given. The aspects listed here are only suggestions. Other aspects can also be relevant, depending on the map purpose.

Initially, a raster image of the vegetation, town and hydrography of the region was integrated in a vector file with transport routes, as shown in Figure 6.12. For this first draft, we make the following observations:

Transport routes:

- Train lines: Including a small gap between train lines can make it easier for users to follow the lines on the map.

- Bus routes: The yellow color for bus routes can be confused with normal streets.
Figure 6.12: A first draft for the study of a new schematic map for ZVV. The map presents smooth simplified transport routes and background containing rivers, vegetation and towns. Labels are all at 45 degrees orientation. Source: Institute of Cartography, ETHZ.
- Train stations: Reduce the size of the white cylinders at link stations, especially at the ‘Hardbrücke’ station.

- Bus stations: The gray color of stations should be changed to another visually more distinguishing color in the context, maybe white or black.

Labels:

- Place labels of all transport routes and stations in the study area. This is important to decide how to fit them.

- It can be enough to place labels for train lines only at the end of routes.

- Regarding the labels of stations, use their sizes according to the importance of the station or to the transport mode. Preferably use only one or two sizes.

Background:

- Rivers: Color not well distinguished in the map.

- Forests: Too many small forest areas, test to include only the ones which can be important for orientation, or use a lighter green on them.

- Streets: Include some main streets and highways in the background.

- Towns: Change color of villages from purple to gray, or use a lighter purple.

- Tunnel: The dotted lines for representing tunnels at the right side of the map may be confusing for travelers. They may interpret them as an event in the transport lines or services, e.g., a special schedule or a longer path. Since tunnels are not a characteristic of the transport lines, but part of the external context, use another representation for them, such as outside marks on the line at the two ends of the tunnel.

- Reference places: Include some reference places in the background map.

A second draft was then prepared including some changes and more elements in a little enlarged map area. The draft is shown in Figure 6.13. Following we comment it.
Figure 6.13: Another draft for the study of a schematic map for ZVV. Highways and streets, as well as more train lines, were included. *Source:* Institute of Cartography, ETHZ.
Transport routes:

- Bus routes: Use a stronger color than yellow for streets with buses, or draw one line inside the streets containing bus routes. These alternatives can facilitate viewing where buses drive.

- Train lines: The yellow color of line S10 may be difficult for some people to be read in the map. Use another color on it (e.g. brown) or a dark frame around the yellow (e.g. black).

Labels:

- Prepare a version with horizontal station labels for comparison.

- It should be clear that small size of letters are for buses and larger size for trains, but, for example, station names of train line S17 have small size.

- Bus lines are not clearly identified.

Background:

- Rivers: Color is confusing with transport lines S3 and S17.

- Forests: Use a lighter green.

- Towns: It is not obvious that purple areas represent villages.

- Streets: Analyse if the streets included are helpful for the purposes of the map.

Others: Linear features in some areas can be stretched to fit stations and labels better, for example, near the station ‘Bergfrieden’.

In Figure 6.14, we present one final proposed new design for the ZVV schematic map.

A raster image of the terrain was included on the background. The highways and streets without bus lines were considered to densify the appearance of the map too much. It was also decided not to include reference places. The representation of tunnels in the first draft was not considered to be confusing. Note that this map looks more like a traditional generalized topographic map, which is intended to represent the real world as faithfully as possible. The background is quite complete and only the bus lines were intentionally more simplified. The information was not graphically controlled for helping the user navigation. We recall that in
Figure 6.14: Proposed new design for the ZVV schematic map. *Source*: Institute of Cartography, ETHZ.
schematic maps, it can be desirable to distort elements beyond the distortions required for representational reasons, in order to omit unnecessary details for transportation, to simplify shapes and structures, and to make the maps more readable for wayfinding tasks. The visual appearance of a schematic map depends not only on the graphic symbology, but also on the interrelationships between all map elements, and on their data position in the map space. The more clearly we relate schematic maps to the critical elements of the represented environment, the more easily users will find a wayfinding solution [Casakin et al., 2000]. More time and work is, therefore, still necessary to make such adaptations for the new schematic map. Yet the project had a short duration.

6.3 Discussion

In the first experiment, we had a database which was used in the automatic schematization of lines with the iterative approach. From such a database, one can derive numerous products either for digital or paper publication purposes. The schematization was achieved in only a few minutes. Most of the time to make a schematic map could be used to analyse the transport system and context, and to add cartographic features and labels to the map visualization, according to the chosen design style.

Although iterative algorithms can be quite time consuming, the tests show that the algorithm performed quite well in terms of time. The total time of the production process of our schematic maps is substantially less than if the mapmaker first had to schematize the lines and still had no information about how to design a cartographic schematic map. Recall that we considered only a relatively small area in our experiment. For the whole transport network of a city or region, there can be hundreds of transport lines to be schematized.

The schematization approach used in this thesis depends on the initial configuration of the road network. The process may be monitored as it progresses, as far as relaxation of constraints is concerned. The convergence problems in our case were satisfactorily solved, because there is not only one possible schematization of a line network. We focused in this approach in preserving the map topology, but there are still other issues that are also desirable to be preserved in the resulting schematization, such as the graphical continuation of main streets and sidedness relations among geographic features.

After the automatic schematization, the diagrammatic graphic still have to be carefully controlled. The best diagrams balance the equally important forms of
information into an aesthetically pleasing whole. The scale and direction relationships can be carefully arranged to create a favorable impression of the transport services of the region.

In the design of a schematic map, it is necessary to choose adequately which elements should appear in the background, and an effective cartographic representation for all map elements. With the information given in Chapter 2, design characteristics for the new map could be quickly decided. Mapmakers can still use their ability to innovate, adding private and personal information to the resulting schematic lines. Recall that a schematic map is partly a representation of reality and in part a product of its maker.

Some users would also like to have an automated design of the route network, including, for example, automatic symbolization for transport modes, or of other events on routes such as traffic volumes and roadway conditions. This would require a symbology that works successfully with no or little human intervention, something which even the most advanced cartographic tools can not offer. However, automated label placement could be included.

In the ZVV project, one professional cartographer worked more than 80 hours on the selected study area. The main design decisions were made after analysing the intermediate test drafts, based on the experience of the cartographer with conventional maps. But choosing the design of a schematic map for a given situation is subject to more criteria than in conventional map settings, notably result quality next to resource consumption and users' comprehension of the map. A poor design can frustrate or confuse the map user. Actually, the effect of a map depends in large part on the user's skill, experience, and perceived needs [Robinson et al., 1995]. Therefore, to be really effective, the map design must also consider its users, their needs, and their perceptions. The map designer's function, in the case of schematic maps, is not to supply a quick fix but to be prepared to track a long way - what might appear at first to be a nice, stylish cartographic design in the opinion of the mapmaker, may not be comprehensible enough for the large number of users of a schematic map for transportation.
Chapter 7

Conclusions

7.1 Summary

Schematic maps, such as subway or transit maps, are produced by hand or using graphic software at present. This is not only a time consuming process, but requires a skilled map designer. Automatic generation of schematic maps may improve the process, but more importantly, would extend the use of such maps to a larger audience.

As a first contribution, this thesis identifies cartographic aspects that are required to be taken into account to generate good designs of schematic maps. There is still a lack of investigation and methodology for the creation of schematic maps. The knowledge from route-based mapping depends in part on the ability of cartographers to define the relationship between route data and visual representations.

In order to produce schematic maps automatically, we need a database containing geographical and topological information about the transportation of the city or region under consideration. Currently, most existent data models are not able to handle fundamental elements of transport network as required in our application [Nielsen et al., 1997]. We have provided a conceptual data model which reflects the requirements of our database: to be used as input in the automatic generation of schematic maps and as a general information system to users of transport networks. The data model successfully address user queries regarding location of stations, adjacency of stations, proximity of stations to reference places, and crossing of routes and rivers. We have also provided a framework to interact with electronic schematic maps based on this data model. We implemented a prototype tool which visualizes the transport routes and supports user location queries.
The generation and visualization of schematic maps on demand is approached in this thesis as a geometric optimization problem. Previous work on the automatization of schematic maps focused mainly on the simplification of lines, without considering other design characteristics of schematic maps [Neyer, 1999, Barkowsky et al., 2000]. In [Elroi, 1991] and [Cabello et al., 2001] simplified lines have fixed directions, but in the first approach no technicalities are given about how to solve topological conflicts among linear features, and in the second one it can happen that no schematization is found at all. Our schematization process modifies the original road network based on common sense manual displacement constraints used in many existent schematic maps. The constraints we consider are straight lines, fixed directions, minimum distance between pairs of map elements, and maximum length for line segments. Because the constraints can be contradictory, the iterative method is used to find a compromise solution. The difficult aspect in repositioning the route network is, though, to preserve the map topology. Our method does handle spatial conflicts in the displacement process. We propose an algorithm which preserves the topological properties of the route network and prove to be useful in practice.

Despite the fact that iterative methods do not guarantee to find an optimal solution, this choice was appropriate since the approach has large potential in this area and it has shown good results. But there is a tradeoff that must be considered in any heuristic algorithm. If the neighborhoods to search are too large, and the emphasis is on speed, iterative methods could not be appropriate.

As a final contribution, this thesis regards quality of schematic results. We have analysed the iterative method and found constraints depicting the length, angle, and shape of roads for evaluating quality of our results. With optimization techniques, getting the software to achieve results comparable to a human relies to a large extent on being able to quantify the criteria used in the applied solutions, in order to evaluate the acceptability of results. We presented a function that assigned a quality score to each schematic map. The quality function is based on the angular error constraint of line segments. Nevertheless, evaluating mathematically the visual quality of a map is not a trivial task. Although some measures of quality considering legibility issues exist, quality is a holistic concept and it is considered doubtful whether even a cartographer can define a perfect map [Wilkinson, 1987]. We recall that there is not only one solution when producing a schematic map, but there are several useful solutions.
7.2 Discussion

Difficulties encountered

The work and results presented in this thesis have their source in an interdisciplinary collaboration between computer science and cartography. Actually, the study of schematic maps on demand can involve the fields of cartography, computer science and the engineering field of transportation. It is common sense that the advantages of such interdisciplinary work are many. Different disciplines regard a problem differently influenced by their different background, and the sharing of knowledge between the disciplines can open new ways of solving problems.

However, interdisciplinary work can cause some difficulties not necessarily present in a project dedicated to a single field. Each field has a different set of priorities and requires specific knowledge to their comprehension. Added to it, people from different disciplines speak different languages. Consequently, both parties need to be open to understand each other and also to learn to communicate, trying to avoid unspoken assumptions. We also need to consider that more unexpected new requirements can affect the solution offered. It is important we can adapt on the fly with the very best of intentions as new situations present themselves.

Benefits of schematic maps on demand

With the insights that have been offered in this work, we can begin to see how computer-based technology can help us to go beyond sketching routes in assisted methods. We are convinced that the facilities of electronic schematic maps on demand as we proposed can affect positively various aspects of modern life.

The most relevant result yielded by the automatization of the schematization process is to reduce the amount of human effort involved. Through the consistent application of specific constraints, the computer offers the possibility of an initial map, which can be further improved by post-processing. Mapmakers may be given the option of having their own schematic maps, resulting from experimentations with different sets of preferences, layout's fidelity and aesthetical principles in combination with the individual priorities. Furthermore, the method can easily be extended with new constraints. The only requirement is that it must be possible to express the constraints as functions of point displacements.

The true benefit of the work can therefore be realized by means of the incorporation of schematization processes into the proliferating number of GIS and cartographic systems. Schematic maps can be automatically generated to enlarge the
possible alternate outputs for graphic representation of networks in GIS. Schematic representations in GIS can give the user the graphic environment, the database, and the analytical power of GIS, plus the benefits of quicker results.

With the computer generation of schematic maps, mapmakers will be more aware of choices, values, and possibilities. The hand made design of schematic maps have similar goals and biases, however they are vaguely defined and unevenly applied. Automated mapping allows experimentation with different sets of constraints.

Finally, it is clear that schematic maps and displays are not substitutes for spatially accurate and detailed network maps. Schematic maps can facilitate a simpler and faster visual interaction between users and network data.

We hope with the study of schematic maps for transportation that in the future more cities can afford developing and producing schematic maps for their collective transports. This venture could help them to improve their transportation system and to attract people away from their cars. In doing so, they can even contribute to the international achievement of urban transport policy objectives to reduce congestion, offer greater safety, better access, fairer opportunities, enhance environment, improve quality of life, and increase sustainability.

As mapping for linear reference systems develops, mapmakers have an opportunity to consider whether it is to be the techniques, or the mapmakers, that will be making maps. Yet, we recall that in cartography the last decision for the design of a map, also considering digital production methods, stays with the map producer [Hurni, 1995].

### 7.3 Directions for future research

The study of schematic maps presents a new subject in cartography and in automated cartography at present, so, compared to other topics, such as map labeling, it still requires more work. Research into schematic maps on demand can and should proceed in different ways. Some possible extensions of the thesis and directions for future research are on the following issues. The issues go from cartography to computer science aspects:

- New perception and cognition studies could contribute much to the knowledge about visual perception of points and lines that are superimposed on, or offset from, a route network. Perceptual and cognitive research is called for to improve the communication effectiveness and formalization of rules to use
symbologies for route data, for both the print and computer screen environments. This is important, because to be effective, cartographic design must take into account the map users, their needs, and their perceptions.

- Weights can be defined for the constraints considered in the displacement process. These weights could be defined for object types or even for individual objects. Furthermore, the weights of the constraints could be functions of existing attributes in the database, including geometry. This would be a step towards introducing semantic meaning into the displacement behavior [Harrie, 1999]. This could also be applied in the simplification of lines. As the different types of linear entities (i.e., streets and rivers) play different roles in the interpretation of the resulting map, it may also be sensible to use different degrees of simplification for either of them [Barkowsky et al., 2000]. For example, streets may be simplified up to the highest possible degree, whereas rivers may remain curved to a certain extent to convey their characteristics of being natural entities.

- Current approaches for computer schematic mapping maintain a constant scale factor for the entire map. When a constant scale factor is used for the schematization of routes, it may force the shorter roads to shrink to a point and essentially keep influencing the neighborhood. Handdrawn maps make scale distinctions and exaggerate the lengths of shorter roads to ensure that they are visible. Computer cartography should also make the deformation of the map easier to carry out.

- The iterative method only displaces and distorts network lines. It may also be desirable at certain points to distinguish paths coming from other directions to show that there is a natural continuation at that point. The perceptual grouping principle of 'good continuation' [Thomson and Brooks, 2000, Thomson and Richardson, 1999] could serve as the basis for partitioning the network into a set of perceptually continued linear elements.

- At this stage, no work with computational and storage efficiency has been carried out. This is another question that deserves further consideration. There are also techniques of parallel processing that could make the method more efficient to be used in practice, also with files that are too large.
Bibliography


generic data model for linear referencing systems. Research Results Digest, 218:7–
18.

Adaptation of geographic information systems for transportation. NCHRP Re-
port, 359.


TM PocketMedia. Sociedade de Transportes Colectivos do Porto (STCP),

Curriculum Vitae

Education

May 1998 – July 2002  Candidate for the degree of Doctor of Technical Sciences  
Swiss Federal Institute of Technology (ETH), Zurich, Switzerland  
Admission Test: March 1999, Prof. Dr. Niklaus Wirth  
Institute of Theoretical Computer Science, 1998-2000  
Institute of Cartography, 2001-2002  
Supervisor: Prof. Dr. Lorenz Hurni

March 1991 – Apr. 1994  Master of Computer Science  
Department of Computer Science  
Federal University of Minas Gerais, Belo Horizonte, Brazil

March 1987 – Dec. 1990  Bachelor of Computer Science  
Department of Computer Science  
Catholic University of Minas Gerais, Belo Horizonte, Brazil

March 1984 – Dec. 1986  High school in Belo Horizonte, Brazil

Teaching Experience

Swiss Federal Institute of Technology (ETH), Zurich, Switzerland  
Taught Algorithms & Data Structures 1, 2 and 3, Logic for Informatics, and Programming of Software Tools 2.

Aug. 1995 – Apr. 1998  Assistant Professor of Cartography and Geoprocessing  
Dept. of Cartography, Federal University of Minas Gerais, Brazil / Taught Cartography, and GIS.
Aug. 1992 – June 1993  Lecturer at the Dept. of Computer Science
Federal University of Minas Gerais, Brazil
Taught Computer Programming for Engineering.

Technical Experience

Apr. – May 1995  Specialized Technician for United Nations Development Program (UNDP) / Worked on the implantation of commercial Internet in Brazil.

May 1994 – May 1996  Researcher, Brazilian Research Network (RNP)
Worked on projects for dissemination of Internet in Brazil.


Apr. 1989 – June 1990  Work Term Student, Telecommunications of Minas Gerais
Computer programmer.

Distinctions


• Graduate Research Fellowship from National Council for Scientific and Technological Development (CNPq) at the Brazilian Research Network, May 1994 – Apr. 1996.

• Master’s Fellowship from ‘Coordenação de Aperfeiçoamento de Pessoal de Nível Superior’ (CAPES), March 1991 – March 1993.

• Medal for student with most outstanding grade in Computer Science (46 students), Dec. 1990.