INAUGURAL - DISSERTATION

zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich-Mathematischen
Gesamtfakultät
der
Ruprecht-Karls-Universität
Heidelberg

Vorgelegt von
Diplom-Mathematiker, Systemprogrammierer Maksim Rylov
aus: Bischkek, Kirgisische Republik

Tag der mündlichen Prüfung: 25.11.2014

Gedruckt mit Unterstützung des Deutschen Akademischen Austauschdienstes.

Thema

Models and Methods for Cartographic Label Placement

Gutachter: Prof. Dr. Alexander Zipf
Prof. Dr. Jan-Henrik Haunert

This dissertation is dedicated to
Daria, Veronika and Daniil

Acknowledgements

First of all, I would like to thank my supervisor and mentor Prof. Dr. Alexander Zipf for giving me a chance of being a part of the GIScience research group and to pursue my doctoral degree at Heidelberg University. I highly appreciate his help, guidance and support over a long period of time when I was working on my dissertation. Furthermore, I would love to mention his assistance and care in providing a great working environment.

I wish to thank Jun.-Prof. Dr. Bernhard Höfle for actually being my mentor. I appreciate him for careful reading and reviewing of my annual reports, for conducting the journal club which helped me a lot to gain my first experience in the methodology of writing and reviewing a scientific paper.

I especially wish to warmly thank my colleague and co-author Andreas Reimer for many interesting, inspiring and productive discussions. All our research papers have benefited from his profound knowledge of traditional cartography.

Additionally, I particularly grateful to my present and former colleagues, especially to René Westerholt, Pascal Neis, Oliver Roick, Lukas Loos, Enrico Steiger, Johannes Lauer and Timothy Ellersiek for the help, interesting teamwork and fruitful discussions.

I also wish to express particular appreciation to the anonymous reviewers for their constructing and helpful comments and suggestions, who helped to significantly improve the originally submitted papers.

Furthermore, I would like to gratefully acknowledge Deutscher Akademischer Austauschdienst (DAAD, German Academic Exchange Service) for awarding the DAAD-Scholarship and giving me an opportunity to pursue my doctoral degree.

Finally, I wish to thank my parents, as well as my wife Daria and two my children Veronika and Daniil for their endless support, encouragement and infinite patience. Their true love and warm closeness is the most important source of my inspiration.

Abstract

Maps are a medium of communication that conveys geographic information through a graphic representation. Cartography as science and art is being developed over centuries by collecting and refining design principles and postulates for depicting high density graphics. These principles completely comprise a great variety of tasks that arise in map-making. One of the essential parts of any map is the lettering. The lettering is a process of locating and assigning names to geographic features such as points, lines or areas. A properly conducted positioning of labels aids map reading considerably. It is well known that label placement is a tedious and time-consuming sub-task of map production. For this reason, over several decades people were trying to automate this process. The automation of the lettering process includes all aspects of label placement such as text content, text design and text placement. The problem of text placement received the most meticulous attention from researchers. There have been numerous and various research attempts to address this problem. However, the problem still persists and remains attractively challenging. The greatest challenge of the automation consists in how to teach a computer program to consider as many diverse, informal and contradicting principles of map labelling as used by a human cartographer in one's work.

This dissertation introduces novel models and methods for cartographic lettering. The research supplements the existing knowledge in the field of automated label placement and presents several new methods for some specific cartographic tasks which have been neglected in the preceding research efforts. The main focus of the conducted research is the attempt to bring automated label placement to a qualitatively new level by considering as many cartographic precepts as can be found in the corresponding literature. To be more precise, firstly, the dissertation presents a comprehensive multi-criteria model that complies with almost all known cartographic principles for point-feature label placement. Secondly, a novel generic raster-based model is introduced. This model allows full automation of the refined techniques for improving map feature overlap, visual contrast and layer hierarchy. Each of the proposed models is expressed as a quality evaluation function which can be used by any algorithm that solves a combinatorial optimization problem, i.e. finds an optimal label placement. Thirdly, two new algorithms to automate particular cartographic tasks are developed. These tasks are the lettering of areal features externally and pairwise labelling of linear features that present geographic boundaries.

The results of the conducted research revealed that the meaning and the purpose of map lettering is poorly understood when it comes to automation. This fact conditions careless and unprofessional label placement on the maps produced by means of computers. The proposed models and methods give an instrument to amend the situation and try to decrease the limitations of the related work. However, the developed algorithms include only some part of the diverse tasks used in map labelling. The carried out experiments show that the proposed labelling techniques are superior to existing solutions which, in turn, take less cartographic guidelines into account. The resulting maps involve a much clearer and more legible lettering of geographic objects.

The results of the presented dissertation highlight a high necessity in further research regarding the map labelling problem. In addition, the findings of this research open a new horizon for a further development of automated label placement. Future approaches should be even more sophisticated and more comprehensive in terms of reproducing cartographic principles in corresponding algorithms. In conclusion, the developed methods are described in sufficient details to be replicated either for the purpose of subsequent research or to extend the capabilities of a computer system that deals with map labelling.

Kurzfassung

Karten sind das Kommunikationsmedium zur visuellen Repräsentation geographischer Sachverhalte. Theorie und Praxis der Kartographie haben über die Jahrhunderte einen seinesgleichen suchenden Korpus von Gestaltungsregeln und -maßgaben zur Darstellung hochverdichteter Informationen entwickelt. Dieser Korpus deckt eine Vielzahl von Einzeltätigkeiten und -problemen ab, die bei der ideellen wie dinglichen Kartenproduktion anfallen. Eine der grundlegendsten Teiltätigkeiten ist die Kartenbeschriftung. Das Beschriften der Karte kann als Prozess verstanden werden, in denen punkt-, linen- und flächenhaften geographischen Phänomenen Zusatzinformationen, im überwiegenden Falle Ortsnamen, visuell beigestellt werden. Die zweckmäßige Platzierung der Schriftelemente auf der Karte ist eine der Hauptdeterminanten ihrer Lesbarkeit. Bekanntermaßen handelt es sich bei Schriftplatzierung um eine der repetetiven und zeitintensiveren Teiltätigkeiten des Kartenherstellungsprozesses. Nicht zuletzt deswegen wird seit mehreren Jahrzehnten an automatisierten Lösungen für die Kartenbeschriftungsprobleme gearbeitet. Die Bemühungen konzentrierten sich vor allem auf die Schriftplatzierung, unter einer gewissen Vernachlässigung von anderen Bereichen wie Toponymwahl und Schriftgestaltung bzw. -systematisierung. Trotz der Konzentration vieler Arbeiten auf den scheinbar geometrischsten und somit der Automatisierung am nächsten stehenden Teilbereich, bleibt die Schriftplatzierung auf Karten als Gesamtes eine nicht zufriedenstellend gelöste Herausforderung. Hierbei ist die größte Herausforderung, die Qualität einer maschinell erstellten Lösung mit der von einem Kartographen erstellten Karte vergleichbar werden zu lassen, inclusive aller Gestaltungsregeln in ihrer Ambiguität und potenziellen Widersprüchlichkeit.

Diese Dissertation stellt neue Modelle und Methoden zur kartographischen Schriftplatzierung vor. Die Arbeit baut auf dem bestehenden Wissen zur automatischen Schriftplatzierung auf, und stellt mehrere neue Methoden für eine Reihe von bislang ignorierten Gestaltungsprinzipien vor. Der Fokus der Arbeit liegt in dem Bemühen, das Qualitätsniveau automatischer Lösungen durch die Berücksichtigung aller in der Literatur anzutreffender Gestaltungsregeln signifikant zu erhöhen. Genauer gesagt stellt die Dissertation zunächst ein umfassendes multikriterielles Modell zur Beschriftung von Punktobjekten vor, welches nahezu alle verfügbaren Gestaltungsregeln beachtet. Anschließend wird ein neuartiges rastergestütztes Modell zur Berücksichtigung des Kartenhintergrundes vorgestellt. Dieses Modell erlaubt die vollautomatische Beachtung der Ebenenhierarchie, der Objektüberschneidung und des farblich-visuellen Kontrasts. Für beide Modelle wird eine Bewertungsfunktion vorgestellt, sodass sie unabhängig von einem konkreten Optimierungsverfahren angewandt werden können. Im dritten Teil werden zwei neue Algorithmen zu kartographischen Spezialproblemen vorgestellt. Hierbei handelt es sich um die externe Beschriftung von Polygonen sowie paarweise Beschriftung von Grenzlinien.

Die Ergebnisse zeigen, dass es bislang große Verständnisdefizite im Bezug auf den tatsächlichen Zweck und die Bedeutung in der Automatisierungsforschung gab. Dies erklärt die beobachtbar nachlässige und unprofessionelle Beschriftung gängiger computergenerierter Karten. Die vorgeschlagenen Modelle und Methoden stellen ein Instrument zur Verbesserung der Situation dar und setzen vorhergehende Arbeiten besser in Wert. Es sei angemerkt, dass die in dieser Arbeit vorgestellten Algorithmen nur einige der diversen Teilbereiche der Kartenbeschriftung behandeln. Die dargelegten Experimente zeigen, dass die erarbeiten Ansätze gegenüber vorhandenen Algorithmen überlegen sind. Sie beziehen weit mehr kartographische Richtlinien ein, was zu einer deutlich höheren Lesbarkeit und zu einer potentiell höheren Informationsdichte führt.

Die in dieser Arbeit präsentierten Ergebnisse verdeutlichen den Bedarf weiterer Forschung im Bereich der automatisierten Kartenbeschriftung. Hierzu stellen die präsentierten Methoden einen Startpunkt für potentiell weiterführende Arbeiten dar. Künftige aufbauende Ansätze sollten ein noch stärkeres Augenmerk auf die Einhaltung etablierter kartographischer Richtlinien legen. Darüber hinaus können die in dieser Arbeit vorgestellten Ansätze auch direkt in existierende Software integriert werden. Sämtliche Methoden werden detailliert beschrieben und genügen demnach sowohl der wissenschaftlichen Reproduzierbarkeit als auch einer direkten Einsetzbarkeit in praktischen Systemen.

Contents

List of Figures

List of Tables

Part I.

Synopsis

> *"Poor, sloppy, amateurish type placement is irresponsible; it spoils even the best image and impedes reading."*
> E. Imhof, 1975

> *"No one algorithm seems to be capable of recognizing the many considerations that a skilled human cartographer is capable of making in lettering a map. Thus, while automated type placement has improved dramatically, and rapidly, it is still not 'there' yet, especially for complex small-scale thematic maps."*
> C.H. Wood, 2000

1 Introduction

This chapter presents the problem statement, describes the objectives, enumerates the research questions and briefly gives an overview of the methods that are proposed to conduct the research.

1.1 Motivation

Over centuries, people have used maps as a medium for communicating spatial information. Until the end of 20th century, traditional paper maps were made by highly skilled map-makers and were typically utilized by cartographers, seafarers, travelers or geologists with professional requirements. Nowadays, people in their daily life deal with maps much more often. *Google Maps*[1] for example had 65 million users in February 2012, *MapQuest*[2] 35 million and *Bing Maps*[3] 9 million respectively (New York Times, 2012). This is caused by the rapid and significant progress in information and communication technology, e.g., wireless networks, geo-location systems, mobile devices and various geospatial and location-based services (LBS; Mooney and Corcoran, 2012; Neis and Zipf, 2008). Paper maps have been replaced by computer-generated maps in various fields. The use of the internet for delivering digital maps (van Elzakker, 2000; Kraak, 2001) can be considered as an important and remarkable transition in the development of cartography (Kraak and Brown, 2001; Peterson, 2003). Progress brought to mankind new technologies such as Web Services (Peterson, 2012): *Web Map Service* (WMS; OGC, 2006),

---

[1] http://maps.google.com

[2] http://www.mapquest.com

[3] http://www.bing.com/maps

Web Map Tile Service (WMTS; OGC, 2010; García et al., 2012), *Web Coverage Service* (WCS), *Web Feature Service* (WFS), etc. The unprecedented growth of different cartographic web services (*Google Maps*, *Nokia Maps*, *Bing Maps*, *Navteq*, *Mapbox*, etc.) and the availability of a huge variety of user-defined maps (Peterson, 2008) have expanded the distribution of maps. This process facilitated an acquaintance of ordinary internet users with modern web cartography (Plewe, 2007; Tsou, 2011). Moreover, users have immediately become participants of the map production process. During the last decade, we witnessed the emergence and the rapid growth of so-called 'volunteered geographic information' (VGI; Goodchild, 2007). The idea behind VGI is to collect and disseminate geospatial data harvested by volunteers. The popularity of VGI is reflected in such projects as *OpenStreetMap*[4] (OSM; Haklay and Weber, 2008; Ramm et al., 2010), *Wikimapia*[5] and *Google Map Maker*[6]. As a result, "maps can now be created and used by any individual with modest computing skills..." (Gartner et al., 2007). The new era of ubiquitous cartography (Gartner et al., 2007; Peterson, 2008) has been introduced.

Web and ubiquitous cartography are the components of the more general concept of automated cartography, which already counts over fifty years of research and development. As far back as 1959, Tobler discussed the possibility of using the power of computing machines to perform many of the tasks in cartography. Since then, ample research has been conducted in this field leading to many promising results and achievements. However, according to a statement of Mackaness (2006) "...cartography is a poorly understood art and science" and he continues to say that "when it came to automation we failed to develop methodologies that captured the process of design (the idea of creating, and evaluating different cartographic solutions)." This statement also refers to automated label placement, which is an assignment of names to corresponding geographic features (Figure 1.1) using computational methods. Feature annotation is an essential part both in traditional (Keates, 1973; Robinson et al., 1995) and automated cartography. Starting at the work by Yoeli (1972), there have been numerous and various research attempts (Kern and Brewer, 2008) to automate the process of positioning names on maps (see extensive bibliography maintained by Wolff and Strijk, 2009). It is estimated that lettering can take up to 50% of the total map production time (Morrison, 1980). The main reason of automation is to reduce the cost of manual attachment of labels to map objects. Label placement algorithms have matured from being only able to solve the simplest problems (Yoeli, 1972; Basoglu, 1982; Hirsch, 1982) to becoming complex and sophisticated tools (e.g., ESRI's *Maplex Label Engine* (2009), Maptext's *Label-EZ* (2014), Evermap's *EverName* (2014), etc.) that are used in map production. It is worth noting that these imple-

---

[4] http://www.openstreetmap.org

[5] http://wikimapia.org

[6] http://www.google.com/mapmaker

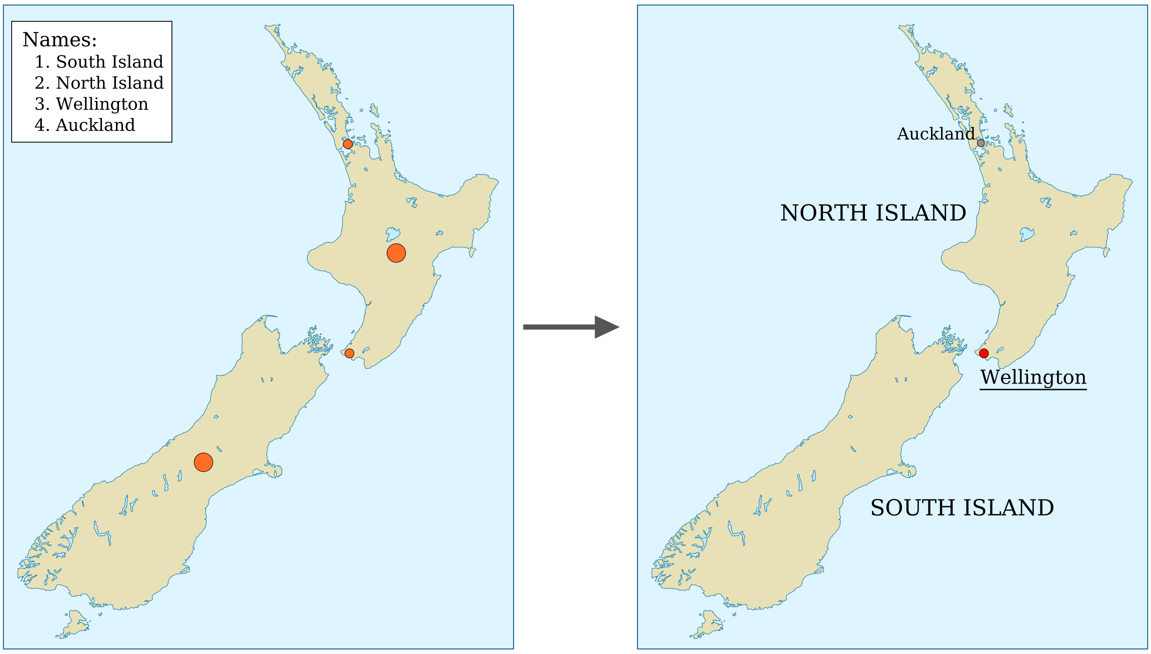


Figure 1.1. The problem of automated label placement.

mentations of labelling algorithms produce rather poor, sparse and amateurish labelling in comparison to lettering by a human cartographer. The reports of National Mapping Agencies (NMAs) such as the USGS (Raposo et al., 2013), Dutch Kadaster (van Altena et al., 2013), and the Ordnance Survey (Revell et al., 2011; Regnauld et al., 2013) argue that map producers have to overcome the limitations of the related work and commercial packages. It is very likely that the stagnation of the research and development of more efficient labelling algorithms is caused by the trends of automated cartography (Gartner et al., 2007) as a part of web cartography. Plewe (2007) noticed that in general, the focus lay on improving the performance of web mapping services (Yang et al., 2005) and their societal implications, as well as on the research on the design of interactive maps (Lobben and Patten, 2003). As a consequence, in spite of the research in this field for over forty years, the problem of automated labelling still persists and remains particularly challenging in several disciplines, such as cartography (Yoeli, 1972; Imhof, 1962/1975; Wood, 2000), Geographic Information Systems (GIS; Freeman, 1991), 3D modelling (Götzelmann et al., 2005; Lehmann and Döllner, 2012) and chart and graph drawing (Battista et al., 1994; Kakoulis and Tollis, 2003). Not without reason it is denoted in the ACM Computational Geometry Impact Task Force report (Chazelle et al., 1999) that label placement is an important research area. A year later, the cartographer Wood (2000) stated, “No one algorithm seems to be capable of recognizing the many considerations that a skilled human cartographer is capable of making in lettering a map.” Furthermore, he highlights that although “...automated type placement has improved dra-

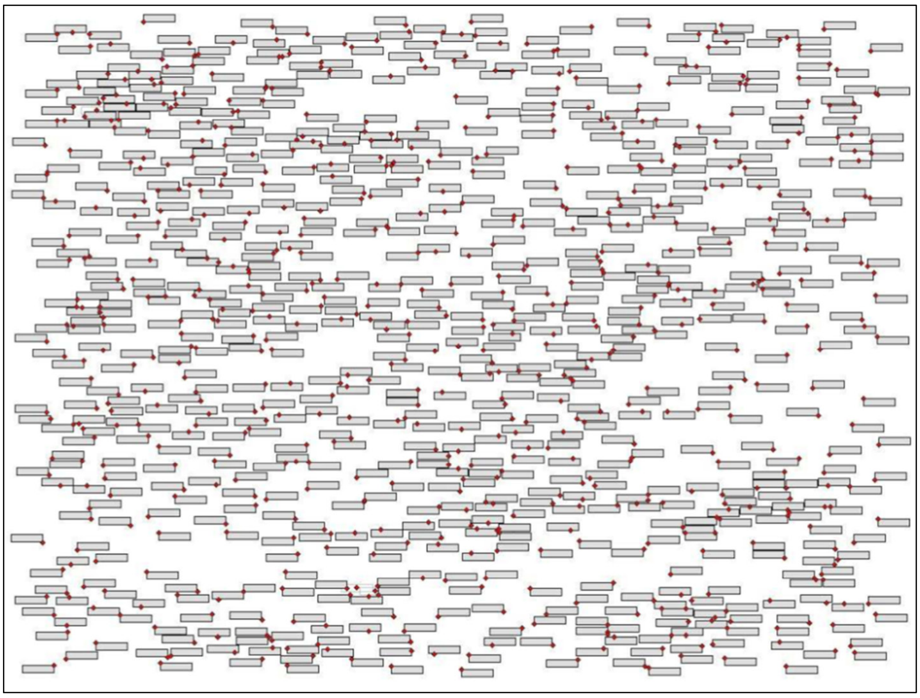


Figure 1.2. Point-feature label placement produced by the algorithm introduced in Ebner et al., (2003). This solution omits almost all cartographic guidelines except one – "avoid label overlaps."

matically, and rapidly, it is still not 'there' yet, especially for complex small-scale thematic maps." Wood's assertion, made almost fifteen years ago, still holds true. The problem is caused by two major factors. First, the map labelling problem has been defined and treated by many researchers as a pure problem of *computational geometry* (for example, Agarwal et al., 1998; Wagner et al., 2001; Ebner et al., 2003; see Figure 1.2) or as a special case of the traditional Maximum Independent Set Problem (Klau and Mutzel, 2003; Ribeiro et al., 2011). Second, many previous attempts dealt with an inefficient formalization of cartographic postulates which can be found in broad and well-established guidelines written down and satisfactorily explained in the literature (Imhof, 1962/1975; Wood, 2000; Brewer, 2005) by cartographers. These guidelines refer to three classes of label placement tasks that are identified in cartography (Imhof, 1972; Keates, 1973; Robinson et al., 1995):

- labelling of point-like objects (e.g., settlements, mountain peaks, POIs);
- linear features (e.g., rivers, boundaries, roads);
- areal features (e.g., countries, islands, lakes, woods);

Despite of many research attempts (Kern and Brewer, 2008; Wolff and Strijk, 2009) to automate these tasks, there are some cartographic techniques used in map lettering that have never been automated yet. Therefore, further research to develop algorithms, which can replicate these techniques, is needed. Furthermore, it would be certainly interesting to understand why existing methods are not capable to produce label placement of superior cartographic quality which is comparable to hand-drawn maps.

1.2 Research Methods and Objectives

The objective of the conducted research is the development of methods for automated label placement in the production of maps. The contribution of the presented research incorporates the enhancement of the existing methods and consists of the creation of novel labelling models and algorithms which cannot be found in the preceding works. These algorithms introduce an attempt to automate some tasks that are used by expert cartographers in the process of map lettering. A detailed description of each research objective is given below.

1.2.1 A Comprehensive Model for Point-Feature Label Placement

Over the last decades there have been many research attempts to solve the problem of point-feature label placement (PFLP). The PFLP problem requires placement of labels adjacent to point features in such a way that overlap of labels is minimized or equals to zero. Different researchers (Kato and Imai, 1988; Formann and Wagner, 1991; Marks and Shieber, 1991) have also proved that PFLP is an *NP-hard* problem. Various compelling techniques have been invented to reduce the PFLP problem such as a *depth-first search* approach (Yoeli, 1972), a *discrete gradient descent* method (Hirsch, 1982), a variant of *0-1 integer programming* (Cromley, 1985; Zoraster, 1986/1990), *exhaustive search* algorithms (Ahn and Freeman, 1984; Freeman and Ahn, 1987; Jones, 1989), *simulated annealing* algorithm (Christensen et al., 1995; Zoraster, 1997), *genetic* algorithms (Verner et al., 1997; van Dijk, 2001/2004; Lorena and Furtado, 2001), a *tabu search* heuristic (Yamamoto et al., 2002) or *artificial intelligence* procedures (Johnson and Basoglu, 1989; Schreyer and Raidl, 2002). In most of the mentioned solutions a quality evaluation function has been used. The quality function computes a single numerical score by using a function that is normally expressed through a weighted sum of single metrics (van Dijk et al., 2002; Hong et al., 2005). These metrics reflect, in one way or another, the formalized cartographic precepts (Imhof, 1962/1975; Wood, 2000). According to van Dijk et al. (2002), this function can be expressed as:

$$\begin{split} Q(L,F) = \sum_{l \in L} \big(w_1 \cdot q_{\text{priority}}(l) + w_2 \cdot q_{\text{aesthetics}}(l) + \\ + w_3 \cdot q_{\text{association}}(l,L,F) + w_4 \cdot q_{\text{label-visibility}}(l,L,F) + w_5 \cdot q_{\text{feat-visibility}}(l,L,F) \big) \end{split} \tag{1.1}$$

where L is a set of labels on the map, F is a set of non-textual map features, w_i, $i = 1,2,\ldots,5$ are the weights. The functions q_*, or metrics, correspond to cartographic criteria summarized by the cartographers in various textbooks and scientific articles. The weights control the contribution of each metric to the overall quality value.

The quality function achieves two main goals: to provide a numerical evaluation of the cartographic guidelines and to compare how well different labelling algorithms perform their task (Christensen et al., 1995). However, most previous research efforts have been focused on point-feature labelling as a geometric problem, where they often omitted extremely important aspects such as aesthetics, label-feature association and feature visibility. These aspects are often ignored or exploited incorrectly. It is worth noting that none of the existing methods can consider all relevant cartographic requirements for PFLP simultaneously. Commercial tools such as Maplex (2009), Label-EZ (2014), etc., which comprise mature and sophisticated, however unpublished and internally closed methods, are limited in their expressive power as they do not take all cartographic guidelines into account either.

Therefore, the first research objective of the dissertation is to construct a comprehensive multi-criteria model which encapsulates most cartographic requirements and principles for PLFP (see Chapter 5), more than any previous model found in the literature. The proposed model should be expressed as a quality evaluation function, similar to equation (1.1), which could be used by any algorithm that solves a combinatorial optimization problem (Schrijver, 2003). The model itself should be highly adjustable and provide human cartographers a handy tool for making an appropriate labelling according to one's preferences. The primary objective of this research is to demonstrate the possibility to considerably enhance the cartographic quality of label placement in comparison to the preceding research. The secondary objective is to investigate computational requirements of the proposed comprehensive model and to understand the contribution of each metric to the overall runtime of the labelling algorithm.

It is planned to test the proposed model on volunteered geographic data (VGI; Goodchild, 2007) provided by the *OpenStreetMap* project (OSM; Haklay and Weber, 2008; Ramm et al., 2010) and also to examine its behavior using different heuristic search algorithms (Christensen et al., 1995) and parameter settings of the model.

1.2.2 Labelling Area Features Outside the Boundary

Lettering of areal or surface features (e.g., countries, lakes, woods, glaciers, islands and island groups, etc.) is one important category of name designations in cartography. By performing this task a cartographer should strive to locate a name by considering both the shape and the extent of the area. A more detailed list of principles for labelling areal features can be found in the works by Imhof (1962/1975), Wood (2000) and Brewer

(2005). One important aspect of labels in mapped areas is that they can be placed both inside the specific feature but also externally, hence outside its boundary. In the field of automated cartography this issue has also received much attention, as it is a challenging problem due to the possible complexity and great variety of shapes. Various algorithms for this task have been proposed by Carstensen (1987), van Roessel (1989), Pinto and Freeman (1996), Barrault (2001) and Dörschlag et al. (2003). However, all foregoing methods present solutions for placing names inside the areas. As far as it is known, an algorithm for the external annotation of areal features is not presented in the literature. Within the range of existing tools for the purpose of generating cartographic output, some proprietary software is available that provide functions for labelling areas externally (Table 6.1). These toolkits are the advanced commercial packages such as Maplex (2009) and Label-EZ (2014), and their description and source code are not available. However, analysis of the preceding work and the investigation of the capabilities of the toolkits revealed their inefficiency in emphasizing the feature-label relationship, which is the most important cartographic guideline for this task. The adduced argument demonstrates the necessity of the development of an efficient algorithm for a given cartographic task.

The second research objective aims to develop an algorithm which performs labelling of areas outside their boundaries (Chapter 6). The algorithm should be able to handle as many cartographic precepts for a certain type of feature as possible. Moreover, the devised algorithm should consist of two procedures: generation of label positions and their quality scoring. Therefore, the algorithm can potentially be utilized by any general labelling algorithm (Edmondson et al., 1996) used either in a desktop GIS application (e.g., QGIS[7], MapInfo[8], etc.) or in a web map server (e.g., GeoServer[9], MapServer[10], etc.). In order to demonstrate the efficiency and to detect possible shortcomings of the presented algorithm, it is planned to draw a visual comparison between its output and a label placement produced by Maplex (2009) and Label-EZ (2014).

1.2.3 Considering Basemap Detail in Label Placement

In many cases, topographic maps feature highly dense graphical information, especially for the case of annotation. Cartography as science and practice has a wealth of guidelines and design techniques to present a combination of visual elements in a clear and legible way. Readability and legibility of a map directly affects the perceptual and cognitive process used by map readers to search names on the map and to recognize their mean-

---

[7] http://www.qgis.org

[8] http://www.mapinfo.com

[9] http://geoserver.org

[10] http://mapserver.org

ings (Lloyd, 1997; Noyes, 1980). Brown (1976) and Phillips et al. (1982) found that among other factors, map background, visual clutter and number of distractors strongly influence visual search and reaction time. For the task of map lettering cartographers use two design solutions; firstly to place toponyms in regions with lower graphic complexity (Castner and Eastman, 1985) and secondly to locate labels in areas where feature and ground differentiation in contrast is higher (Wood, 1994). Research regarding the problem of overprinting other non-textual map features with labels has been also conducted in the field of automated label placement (Jones, 1989; Harrie et al., 2004; Zhang and Harrie, 2006). But unfortunately, it has received insufficient attention resulting not only from the lack of map background characteristics being considered but also from the inflexibility of the proposed approaches to be used in other methods supporting different feature types (i.e., points, lines and polygons).

The third research objective (Chapter 7) focuses on defining an exhaustive quality measure, which uses raster map representation for scoring potential label positions in terms of the amount of visual clutter (Phillips and Noyes, 1982) and overlapping other map elements. The designed measure should take the extent, the shape and the importance of map background features into account. Furthermore, it is of high importance to address the fields of applicability of the model and to draw a comparison of maps that are labelled using this model and an implementation of the vector-based approach (Strijk and van Kreveld, 2002). This comparison should span three aspects such as labelling quality, computational requirements and performance. Furthermore, the devised model should help to partially answer the following questions: What are merits and flaws of a raster-based and a vector-based approach? Can the devised model be used for annotating maps whose background consists of non-geometric features like in the case of 3D models (Götzelmann et al., 2005; Lehmann and Döllner, 2012) or hill-shading (Imhof 1982/2007; Jenny and Hurni, 2006)?

1.2.4 A Method for Annotation of Geographic Boundaries

Over the centuries cartographers were collecting and inheriting distinct precepts for producing good label placement on maps. It is known that these rules are formulated for three types of designations (Imhof, 1962/1975). They are position (e.g., settlements, mountain peaks), linear (e.g., roads, rivers, boundaries) and areal (e.g., countries, lakes, islands, glaciers) designations. Each of these categories has a variety of special cases which require a particular treatment and technique for annotation (Wood, 2000; Brewer, 2005). For instance, linear features that demarcate area boundaries can be labeled using two toponyms, i.e. one toponym on each side. With this approach a reader can identify regions that lie on opposite sides of a boundary line without difficulty. This kind of pairwise labelling has a main visual advantage, i.e. a map reader is informed about the exact nature of the line, not only its general type. This facilitates to easily distinguish boundaries among other linear objects and amplifies the precise graphic relation between the labels and the relevant map features. It should be noted that two techniques are very prevalent in cartography to perform this task. They are:

- Label is positioned along an imaginary straight line.
- Label is curved following the direction of the polyline.

The algorithms for line labelling presented in the literature (Barrault and Lecordix, 1995; Edmondson et al., 1996; Chirié, 2000, Wolff et al., 2001) do not consider the special case of pairwise line labelling. However, it is worth noting that automated methods for this task do exist, which are currently only implemented in commercial toolkits and engines such as Maplex (2009) or Google Maps. The motivation for this research is to develop an algorithm which not only has a similar functionality but also is described in sufficient details to be used by any other labelling engine that needs a comparable labelling feature.

The fourth research objective is to develop an algorithm for pairwise line labelling to annotate linear features that differentiate administrative divisions or other geographic subdivisions (Chapter 8). The first task for this research is to review cartographic literature to find a list of cartographic principles that are exploited to accomplish this task in manual lettering. The proposed algorithm should achieve two goals. It has to generate candidate positions and evaluate their quality according to the predefined set of specific cartographic rules. As usual, it is required that the algorithm should be defined in the form of being able to potentially extend capabilities of any labelling toolkit and to be used with any general labelling algorithm (Edmondson et al., 1996).

1.2.5 Datasets for the Experiments

Experiments are an important part of any conducted research as they help to test newly developed methods and understand their behavior in different situations. Datasets which satisfy five criteria have been chosen for the experiments:

- coverage of large areas (countrywide or even global);
- richness of semantic information (e.g., population and administrative status of a city);
- variety of feature types (e.g., points, lines, polygons);
- appropriate scale of data for small and medium scales;
- availability of tools for the data pre-processing.

Therefore, datasets which suit these criteria will be utilized. The following data sources were opted among all diverse publicly available datasets:

- *OpenStreetMap* (OSM) represents free, crowdsourcing, geo-spatial data gathered by the contributors (OSM; Haklay and Weber, 2008; Ramm et al., 2010).
- *SRTM* (Shuttle Radar Topography Mission) provides elevation data on a near-global scale (over 80%). The SRTM data of version 4 is available as 3 arc second (approx. 90m resolution) digital elevation models.
- *ASTER* – digital elevation model that was collected using Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER GDEM, 2011). The ASTER GDEM version 2 has 30m resolution.

OSM data were used in all research papers (Chapters 5-8). ASTER GDEM have been exploited in the paper given in Chapter 7. SRTM data have been utilized for relief representation and labelling of a world map which is available online through the *OpenMapSurfer* (2014) service (Section 2.5.2).

1.3 Research Questions and Objectives

The main objective of this research is to approach automated label placement in a way to produce labelling which is comparable to lettering positioned by a human cartographer. More precisely, this dissertation aims to introduce new aspects in automation and formalization of cartographic knowledge for annotation of various feature types. The presented work has two main focuses. First, it tries to complete existing approaches and methods towards more readable and legible labelling. Second, it presents some novel algorithms to fill the gap between diverse design techniques used in manual cartography and their counterparts in automated labelling.

The primary objective of the dissertation is to answer the following research questions:

- Which design techniques that are widely used in manual lettering have not been automated yet?
- How can we approach automated labelling to comply with more well-defined cartographic principles and requirements for labelling different feature types?
- What are the reasons of large differences in the quality of labelling produced by means of automated methods and lettering that is placed manually by a human cartographer?
- What role does the consideration of cartographic knowledge in the development of automated methods for map labelling play?
- What are special requirements and technical limitations for using more advanced labelling algorithms in the context of their application in web cartography?

The above-mentioned research questions are used to define the list of concrete objectives and tasks:

- To explore the cartographic guidelines related to labelling. To disclose how well these guidelines are automated both in the methods described in the literature and in the algorithms implemented in different software packages.
- To construct a comprehensive multi-criteria model that complies with almost all well-defined cartographic principles for point-feature label placement.
- To propose and to evaluate a generic quality measure that allows full automation of refined cartographic principles such as map feature overlap, visual clutter and layer hierarchy using a raster-based approach to retrieve map background information.
- To develop a fast and highly efficient algorithm that uses both rich cartographic guidelines and algorithms from the field of computational geometry to label area features outside their boundaries.
- To develop a practical algorithm for pairwise lettering of lines presenting geographic boundaries (e.g., international borders, municipal divisions, grid-zones, military zonings, etc.).
- To develop and implement a general labelling framework that embodies all proposed algorithms as a single whole. To use this framework to annotate a map for the whole globe at several scales.
- To test the developed algorithms in the context of web maps (Kraak, 2001).

The workflow which shows the place and the contribution of the proposed methods and algorithms within/for the field of automated label placement and its applications is depicted in Figure 1.3.

1.4 Dissertation Outline

This section presents the structure of the cumulative dissertation by briefly providing information to the content of each chapter.

1.4.1 Structure

The presented dissertation comprises two major parts: (I) Synopsis and (II) Publications. The first part is devoted to describe the motivation of the research (Chapter 1.1), to present the methods being used (Chapter 1.2), and to give a brief introduction to the research questions, objectives and tasks of the dissertation (Chapter 1.3). The research findings, the experimental results and the details of the implementation are discussed in Chapter 2. Finally, the dissertation is concluded with the overall analysis of the findings Chapter 3) and with a discussion of open questions and possible directions of future work (Chapter 4).

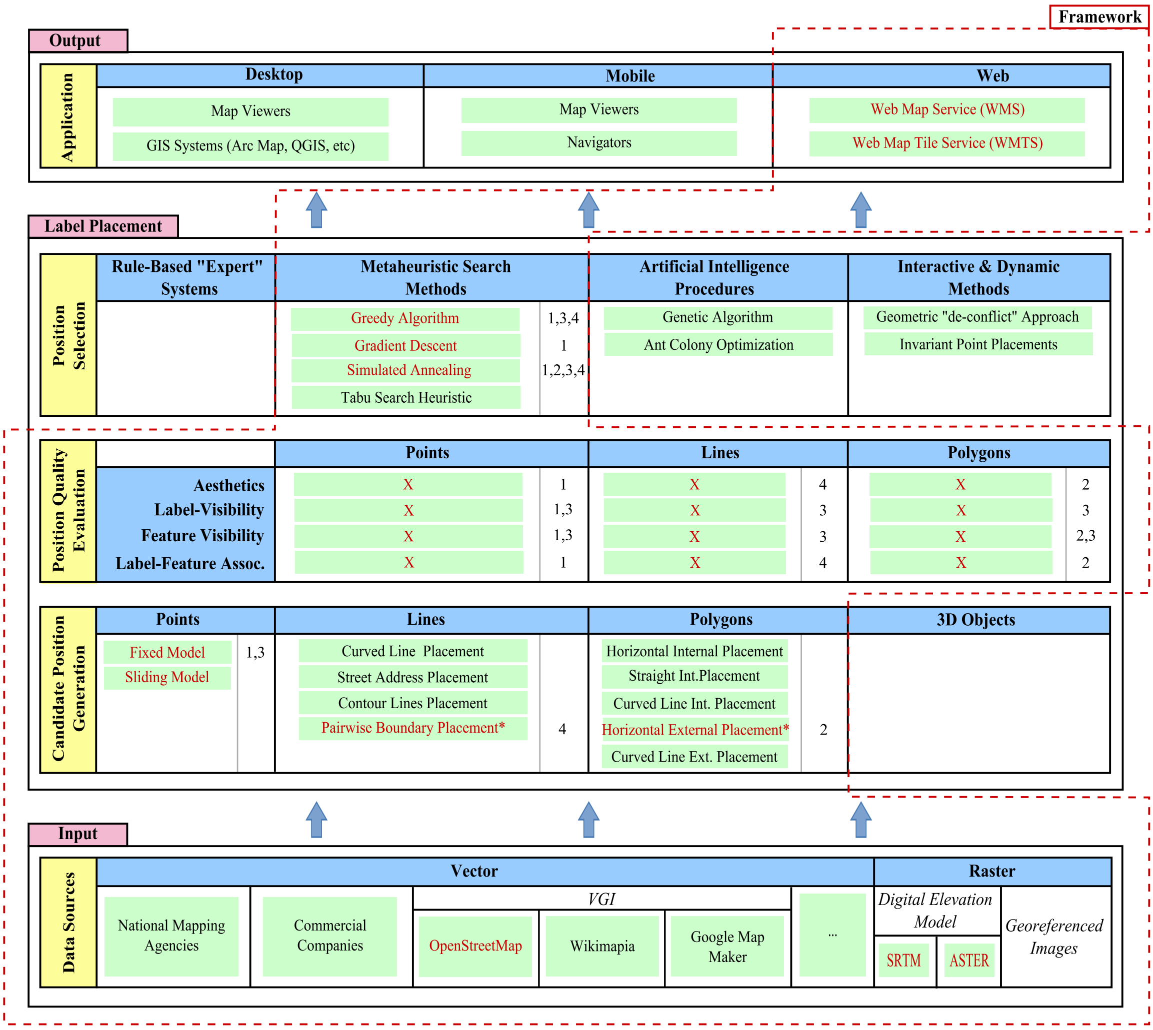


Figure 1.3. The overall workflow of the conducted research. The area within the dashed red line depicts the scope of the dissertation. The numbers show the sequence numbers of the papers (Section 1.4.2).

The second part (Publications) consists of four peer-reviewed articles (see Chapters 5-8, Figure 1.3). The papers deal with the methods of position generation and quality estimation of label placements (van Dijk et al., 2002) for the task of automated map lettering. The proposed approaches extensively rely on cartographic knowledge (Imhof, 1962/1975; Wood, 2000; Brewer, 2005) regarding map labelling and use algorithms from the field of computational geometry (de Berg et al., 2008), linear programming models (Schrijver, 2003), algorithms for solving combinatorial optimization problems (Kirkpatrick et al., 1983; Glover, 1989) and image processing techniques (Pal and Pal, 1993).

Chapters 5, 6 and 8 introduce methods of automation for different tasks of map lettering such as labelling of punctiform, areal and linear features respectively, whereas the model given in Chapter 7 can be considered as a part of the mentioned tasks. To be more specific, in Chapter 5 a comprehensive multi-criteria model that complies with almost all well-defined requirements for points designation used in cartography is proposed. Basically, Chapter 5 presents an attempt to bring the problem of point-feature labelling onto a new level and to make the label placement result relatively close to the one produced by an skilled cartographer. A novel practical method for the external annotation of areal features is given in Chapter 6. This method demonstrates how using a specially designed algorithm for finding candidate positions and a quality measure can produce visually plausible label placements for areas of any shape. Chapter 8 introduces a new efficient and easily configurable algorithm for performing functional pairwise labelling of lines presenting geographic boundaries. Then, Chapter 7 provides a generic quality evaluation model that estimates such characteristics as map feature overlap, visual clutter and layer hierarchy using basemap detail which is given as a raster image. The proposed model is scale-independent and works for any type of designation, i.e., for point-like (Chapter 5), linear (Chapter 8) and areal (Chapter 6) features.

1.4.2 Selected Publications

Maxim Rylov is the lead author of all papers that assemble this dissertation. It is confirmed that he did the major part of the work. His contribution includes statement of the research questions, elaboration and implementation of the methods, and compiling the results for the articles. The co-author Andreas Reimer has contributed to all selected publications by giving constructive comments and suggestions concerned with the raised questions regarding cartographic knowledge. His generous assistance and valuable feedback helped to significantly improve the content, style and quality of the papers as a whole.

The papers are presented in the following sections (Chapter 5-8) and are given in order they have been submitted or accepted for publication in different scientific journals. In order to present the papers in most convenient way, they have been reformatted to a common style.

Publication 1:

Rylov, M.A., and A.W. Reimer. 2014. “A Comprehensive Multi-Criteria Model for High Cartographic Quality Point-Feature Label Placement.” *Cartographica* 49(1): 52-68. doi: 10.3138/carto.49.1.2137

Publication 2:

Rylov, M.A., and A.W. Reimer. 2014. “A Practical Algorithm for the External Annotation of Area Features.” *Cartographic Journal.* doi: 10.1179/1743277414Y.0000000091

Publication 3:

Rylov, M.A., and A.W. Reimer. 2014. “Improving Label Placement Quality by considering Basemap Detail with a Raster-Based Approach.” *GeoInformatica.* doi: 10.1007/s10707-014-0214-6

Publication 4:

Rylov, M.A., and A.W. Reimer. 2014. “Pairwise Line Labelling of Geographic Boundaries: An Efficient and Practical Algorithm.” *Cartographic Perspectives* (submitted)

1.4.3 Additional Publications

Rylov M.A., and A. Zipf. 2012. “Solutions for Limitations in Label Placement in OGC Symbology Encoding (SE) Specification.” In *Proc. Geoinformatik 2012.* Braunschweig, March 2012.

2 Results and Discussions

The purpose of this chapter is to present the results of each part of the study described in a corresponding peer-reviewed article (Section 1.4.2). Furthermore, each following section provides concluding remarks and discusses future research questions that result from the findings.

2.1 A Multi-Criteria Model for the PFLP

The subsections below state the results of the research presented in the following article:

Rylov, M.A., and A.W. Reimer. 2014. “A Comprehensive Multi-Criteria Model for High Cartographic Quality Point-Feature Label Placement.” *Cartographica* 49(1): 52-68. doi: 10.3138/carto.49.1.2137

2.1.1 Results

The problem of point-feature label placement has been considered as a first study (Chapter 5) of the conducted research. It has been demonstrated in Table 5.1 that the PFLP problem was mostly treated as a geometric problem in the preceding research (Klau and Mutzel, 2003; Ribeiro et al., 2011). Thus, it was solved in a way, which disregards and omits many important guidelines used in traditional cartography. Chapter 5 describes a comprehensive multi-criteria model which shows that the theoretical development for solving the PFLP problem is ready, but not completed yet to approach the quality of label placement produced by a human cartographer. The most important and significant findings of this part of the research can be conveyed through the following statements:

- The cartographic guidelines, which were not formalized for automation before, have been found and written down (Section 5.3.2).
- Two novel quality metrics have been introduced. They have been formalized and expressed through analytic formulas. The first metric scores the degree of disambiguation (Sections 5.3.4.4) between neighbouring labels and symbols. The second metric measures the degree of the relationship of a settlement to the shore (Section 5.3.4.6). The definition of these metrics has not been considered yet in the current related literature.
- Point-feature label placement has been performed by taking almost all well-established cartographic precepts into account. The model uses more precepts than any previous research effort.

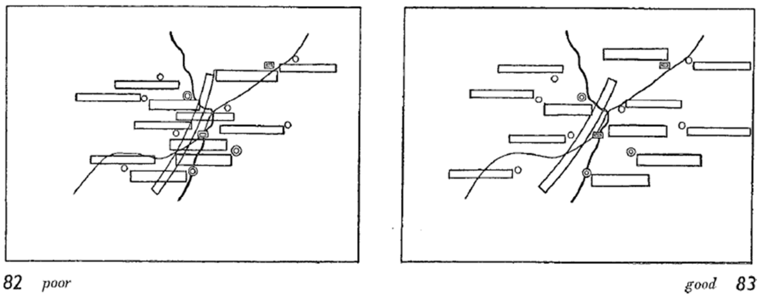


Figure 2.1. An example of poor and good lettering in terms of excessive clustering and ambiguous relationship between nearby labels and their features. (taken from Figures 82-83 in Imhof, 1962, p. 119)

As already mentioned, Section 5.3.4.4 presents a first attempt to address the quantification of ambiguity in label-feature relationships (Figure 2.1; see also Figure 5.8). Remind that an unambiguous relationship between labels and their features is the most crucial issue which a cartographer should strive to achieve. Cartographers such as Imhof (1962/1975), Wood (2000) and Brewer (2005) tried to place a greater emphasis on this issue. The consideration of this issue in the model makes the contribution of the presented research explicitly clear.

Further, it was demonstrated that the developed model is highly adjustable and can be used with any appropriate algorithm for solving a combinatorial optimization problem (Christensen et al., 1995). Being described in sufficient detail, the model can be replicated without difficulties in any labelling toolkit. The conducted experiments showed that the model is capable to produce a visually plausible and, most importantly, functional labelling. The produced labelling can be compared to hand drawn lettering. As expected, in the experiments described in Section 5.4, each quality metric of the model shows its influence on the resulting labelling to some extent. Thus, to obtain a good type placement, a cartographer should parameterize the model carefully.

In summary, it is indispensable to discuss an application of the presented model in web cartography. In order to show the capabilities of the model, a specially designed map has been prepared and published to the web. The map is mostly based on the dataset provided by the *OpenStreetMap* project and covers the whole globe. This map is available on the *OpenMapSurfer* (2014; Section 2.5.2) web page. Furthermore, both map design and lettering are known to be equally important in cartography. Unfortunately, existing developed front-end tools (e.g., ArcGIS Online, CartoDB or Mapbox Studio, etc.) and cartographers, who make use of these tools, aim their efforts at map design, rather than at a carefully and properly located lettering. The importance of lettering is neglected on web maps which have become so ubiquitous nowadays (Gartner et al., 2007). This can be explained by an inefficiency of the back-end frameworks (e.g., Maplex, Mapnik, etc.) to consider as many cartographic postulates as needed for producing a good lettering. For example, point-feature label placement on most online maps is rather poor and untidy. As a proof, the *Map Compare*[11] web tool helps to see and to estimate the difference in the quality of labelling on diverse online maps (see examples in Section 2.5.3) provided by such companies as ESRI, Google, Microsoft, Mapbox, etc. The label placement on "OSM Roads" map, that was annotated using the developed model, overcomes nearly all other maps in terms of adherence to the cartographic precepts (Section 5.3.2) for point-feature labelling (see an example in Figure 2.8). This is especially striking at small and medium scales.

2.1.2 Discussion

The experimental results (Section 5.4) convincingly demonstrated that the proposed multi-criteria model is able to provide a label placement which is very good in addressing cartographic principles. Furthermore, the experiments helped to reveal some rather interesting issues. Namely, the performance of the labelling algorithm highly depends on the number of used quality metrics, as well as on their type. Table 5.3 shows that the increment in the runtime is mostly caused by two metrics. The computation of the metrics $F_{i,j}^{\text{disamb}}$ and $F_{i,j}^{\text{clut}}$ consumes half of the runtime. This can be explained by the nature of an algorithm that was used for searching neighboring labels (Figure 2.1) and for measuring distances between them. A quite straightforward search has been utilized in the implementation. Therefore, some further research is needed to develop a more intelligent and faster algorithm. In addition, a new algorithm for this task should be generic enough to handle the search of labels having an orientation which is different from horizontal. This requirement is governed by the fact that, in theory, the proposed model can be used for labelling linear (Chapter 8; Barrault and Lecordix, 1995; Edmondson et al., 1996; Wolff et al., 2001) or areal objects (van Roessel, 1989; Barrault, 2001; Rylov and Reimer, 2014b). The development of such algorithm can be mainly of interest to computational geometry.

Additionally, when considering cartographic guidelines through a set of quality metrics, the experimental results explicitly indicated a significant slowdown of a labelling algorithm. This is obvious, since any algorithm that solves a combinatorial optimization problem for PFLP is an iterative algorithm. Such an iterative algorithm has an internal iteration that is separated from the main loop. In this internal iteration, each visible label is evaluated through a list of metrics. The greater the sizes of this list, the more number of iterations need to be performed. Therefore, several steps can be taken in order to improve the overall performance. One example is the parallel computation of the metrics. Another example is to use power of graphic processor units (GPU) to speed up a

---

[11] http://mc.bbbike.org

utilized heuristic (Cavuoti et al., 2013; Ferreiro et al., 2013). Moreover, the previous study (Rabello et al., 2013) suggested utilizing a more advanced alternative metaheuristic to find an optimal solution for label placement. Since a metaheuristic plays an essential role in solving map labelling problem, especially when the number of label candidates is large. A large number of candidates can be observed when the sliding model (van Kreveld et al., 1999; Strijk and van Dijk, 2002) is used to generate trial label positions for point-features. Conceivably, all mentioned improvements open up further possibilities to use the presented model for interactive and dynamic labelling (Been et al., 2006; Mote, 2007).

In order to continue discussing possible modifications to the proposed model, it is worth noting that a big set of changes is required to support curved labels, which are used for naming linear (e.g., roads, rivers, etc.) or areal features (e.g., islands, lakes, bays, etc.). First, the metric $F_{i,j}^{\text{pos}}$ should be changed according to a set of cartographic guidelines which are specifically valid for the type of a tagged feature (e.g., point, line, area). Second, the metric $F_{i,j}^{\text{coast}}$ should be omitted, whereas both $F_{i,j}^{\text{disamb}}$ and $F_{i,j}^{\text{clut}}$ require some alterations, as these metrics were designed for horizontally aligned labels.

A raster-based approach to consider overprinting of background features by labels (Section 5.3.4.3) was utilized in the metric $F_{i,j}^{\text{over}}$. This metric has proven its efficiency and robustness. However, it does not consider some very important guidelines that are well-known to cartographers. For example, spatial distribution, importance of background features, as well as difference in brightness (or contrast) between a label and non-textual features also play an important role for a cartographer in making one's decision. Therefore, these guidelines have led to the development of an extension of the raster-based approach that can consider all above-mentioned principles (Chapter 7).

The metric $F_{i,j}^{\text{coast}}$ (Section 5.3.4.6) determines and evaluates in the runtime whether a point-feature belongs to the coast or not. The proposed technique provides some insights and opens up possibilities to develop a method to automatically define locations for labels drawn on the water surface having a curved form (see Figure 2.2 or northwestern part of Sicily in Figure 5.1b). This problem characterizes one identified research gap

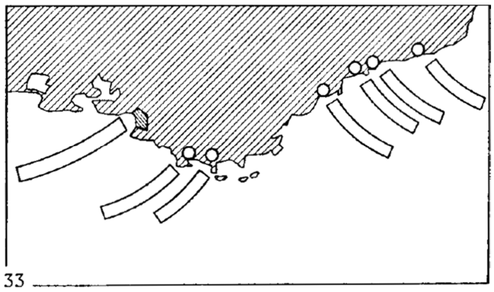


Figure 2.2. Names of coastal places that are curved away from the horizontal orientation. (taken from Figure 33 in Imhof, 1975, p. 133)

which should be addressed in future studies.

Overall, the results of this research highlighted that there are still a number of tasks in automated map labelling which have not been formalized and automated yet. It is hoped that the research findings will give an impulse to construct more practically efficient methods which will be able to achieve superior labelling quality.

2.2 An Algorithm for Labelling Areas outside the Boundary

The following subsections introduce a part of the research that has been published in the article:

Rylov, M.A., and A.W. Reimer. 2014. “A Practical Algorithm for the External Annotation of Area Features.” *Cartographic Journal.* doi: 10.1179/1743277414Y.0000000091

2.2.1 Results

Chapter 6 introduces a novel efficient algorithm for labelling area features outside their boundaries (see its workflow in Figure 6.3). This algorithm comprises two parts that correspond to the problems of label-position generation and evaluation.

Two algorithms were used in the first part. They are the polygon *offsetting* (Section 6.3.2.1; Chen and McMains, 2005; Wein, 2007; Bo, 2010) and a modified version of the *plane sweep* algorithm (Bentley and Ottmann, 1979; Section 6.3.2.2). The modified plane sweep algorithm achieves two goals. First, it creates candidate label positions. Second (Section 6.3.2.4), it determines whether the bounding box of a label intersects the related area or not. Bentley and Ottmann’s (1979) algorithm was used to decrease the number of intersection checks that need to be performed between an input polygon and m rectangles. The algorithm in Section 6.3.2.4 with the runtime $O((n+k)\log n + (k+m)n)$ is not faster for big n than the “naive” algorithm which checks all $4mn$ pairs. However, the proposed algorithm requires less number of operations for small values of n and for polygons that much larger than the size of a label. In practice, the second part of its runtime $O((k+m)n)$ needs less number of operations, since the number of segments that need to be checked for intersection with a label is smaller than n. Generally speaking, to report all intersections between two sets of line segments (a red-blue intersection problem), one can use one of the existing algorithms for this task. One of them is an asymptotically optimal algorithm proposed by Mairson and Stolfi (1988), which reports all those intersections in $O((n+m)\log(n+m)+K)$ time and $O(n+m)$ space, where K is the number of intersections.

The second part introduces a quality measure for evaluating candidate positions in respect to the degree of spatial relationship between a label and its area feature (Section

6.3.3). The experiments showed that the proposed algorithm achieves two goals. First, the quality function demonstrates a rather good performance (Section 6.4.3, Figure 6.10). Second, this measure helps to give preference to label positions which are more desirable (Figure 6.13 in Section 6.4.3) from a cartographic standpoint (Section 6.3.1).

As a result, the developed method produces visually appealing labelling. Furthermore, a comparative study in Section 6.4.4 confirmed that this method outperforms other similar approaches implemented in Maplex and Label-EZ. Since, the developed algorithm incorporates more cartographic precepts than other tested solutions. Furthermore, the algorithm is presented in sufficient detail. Therefore, it can be easily replicated in its entirety in any software package that deals with map labelling. This should be attractive for free and open source GIS packages (Steiniger and Hunter, 2013). Among those are labelling libraries such as *PAL* (2014; Ertz et al., 2009), Geographic Information Systems such as QGIS and uDig, or toolkits such as GeoServer, MapServer and Mapnik.

2.2.2 Discussion

The proposed method incorporates several algorithms from the field of computational geometry (de Berg et al., 2008). This explains the very good performance of the procedure for generating candidate label positions. However, the obtained runtimes have not been compared with runtimes of any other algorithm of similar functionality. Thus, it is impossible to properly conceive the benefits of using those computational geometry algorithms. The problem is caused by a lack of literature and research providing possibilities to benchmark the presented algorithm. Therefore, it is hoped that the well described algorithm can facilitate future development of an even faster or more efficient algorithm.

As many cartographic guidelines (Section 6.3.1) as can be found in the literature were used for the tasks of label-position generation and evaluation. It is most likely that exactly this aspect distinguishes the presented approach from the solutions in Maplex and Label-EZ. The experiments (Sections 6.4.3, 6.4.4) showed that rule R4 (Section 6.3.1), which is obviously neglected in the proprietary toolkits, is highly important for revealing a spatial relationship between labels and corresponding areal features. Moreover, a proper placement of the name around its area has a direct influence on type legibility (Phillips et al., 1977) and on the map reading performance in general (Noyes, 1980; Lloyd, 1997). The proposed proximity measure, based on the Euclidian distance (Section 6.3.3), numerically scores the spatial relationship between a label and its area. This measure has been chosen to prove the initial conjectures. This simple measure with a low computational complexity turned out to be a good and adequate alternative. The evidence of its practical applicability is given in Section 6.4.4. This section presents a visual comparison of label placements produced by the developed method and by two commercial labelling engines. However, the process of how well this proximity measure works has not been investigated to its full extent. For further research, it would be undoubtedly interesting to devise and to develop other proximity measures (Laube et al., 2008; Zighed et al., 2012) and to make a comprehensive comparison between them. Such a comparison should include the following aspects:

- The ability of a measure to reflect a spatial relationship between the name and its area.
- The time needed to find the name for the associated feature.
- The computational complexity of a measure and its effect on the whole runtime of the algorithm.

The results of the experiments are promising. They demonstrated cartographically plausible labelling which fully meets the requirements of the cartographic guidelines (Section 6.3.1). However, this statement is true only for one single area, i.e. an island in the tests. When it comes to label a group of islands which lie very close to each other, another highly important cartographic guideline should be addressed and considered. This additional guideline corresponds to an ambiguous relationship between features and names. Cartographers certainly know that omitting of this principle can spoil any good map. The conducted experiments revealed that, in particular for this labelling task, an ambiguous relationship between names and surrounding features is a crucial aspect and must be considered by a general labelling algorithm (Doerschler and Freeman, 1992; Edmondson et al., 1996). As a part of this conducted research, a first attempt to score an ambiguous relationship between labels and symbols has been made. More precisely, Chapter 5 introduces a model which can perform such scoring tasks, but only for point-features. Hence, further research for the case of "name to polygon" is still needed.

2.3 Quantifying Feature-Overprinting using a Raster-Based Approach

The next subsections present the results of the research given in the following paper:

Rylov, M.A., and A.W. Reimer. 2014. "Improving Label Placement Quality by considering Basemap Detail with a Raster-Based Approach." *GeoInformatica.* doi: 10.1007/s10707-014-0214-6

2.3.1 Results

The third publication (Chapter 7) presents an attempt to develop a generic raster-based model to improve label placement by considering basemap detail (Jones, 1989). The first prototype of the model has been used in a comprehensive model for point-feature label placement (Chapter 5; Rylov and Reimer, 2014a). A draft of the model (Section 5.3.4.3) demonstrated its efficiency in enhancing the overall quality of map lettering (Section 5.4). Remind that only the homogeneity of the map background under a label has been considered in the draft. Next, an overview of the cartographic principles found in the literature (Imhof, 1962/1975; Wood, 2000; Brewer, 2005) additionally detected three essential guidelines (G2-G4, Section 7.3.1) that should be considered in the raster-based method. These new guidelines have encouraged to elaborate three specific quality metrics (Sections 7.3.4-7.3.6) that correspond to the guidelines G2-G4. The metric for G2 (Section 7.3.4) evaluates the spatial distribution of non-textual background objects by using the concept of “entropy” (Shannon and Weaver, 1964; Li and Huang, 2002). The next metric for G4 (Section 7.3.5) takes the distinct importance of background elements into account. The last proposed metric (G3), given in Section 7.3.6, introduces a function that computes the difference in contrast between the colours of a label and the underlying background (Wood, 1994).

The experiments conducted in Section 7.4.3 showed that the refined raster-based measure (Section 7.3.7) is capable to significantly improve type placement in terms of its clarity and legibility. For example, a visual comparison of Figures 7.9 and 7.11 or Figures 7.14 and 7.15 respectively can give an impression of the achieved improvements. The measure proved its consistency and applicability on maps with dense graphical information in the background. Moreover, a visual comparison (Section 7.4) of label placements produced by the developed model and by Maplex demonstrated very much plausible and in most cases similar results (see Table 7.2). However, Maplex has an implementation of the vector-based method (Freeman and Ahn, 1984; Strijk and van Kreveld, 2002). Furthermore, this engine uses the *sliding* model (van Kreveld et al., 1999; Strijk and van Kreveld, 2002) and, according to ESRI’s White Paper (2009), generates exactly 96 candidate positions around every point-feature. Conversely, the *fixed position* model with eight standard positions (Figure 7.8) was used in the developed implementation. Hence, the presented method has a potential to improve its label placement. By operating with 96 label-positions for each point-feature, it will have much more freedom in avoiding overlapping or concealment of background features.

In summary, the developed method has proved to be applicable for a given task and promising for being used in practice. A demonstration on how sufficient the algorithm works can be seen in the provided sample maps depicted in Figures 7.11 and 7.15.

2.3.2 Discussion

The developed measure utilizes a technique of image segmentation (Section 7.3.2; Haralick and Shapiro, 1985; Pal and Pal, 1993). A segmented representation of an image helps to perform an efficient and meaningful analysis of the properties of a raster image.

The experiments (Section 7.4.3), involving *octree* quantization algorithm (Gervautz and Purgathofer, 1988) for image segmentation, have proven the feasibility of the proposed approach. However, some very important questions regarding image segmentation are still open and require further investigations. These questions can be formulated as follows:

- Which image segmentation algorithm is better for which particular type of map (e.g., roads, bathymetry, geological structures (Lisle, 2003), shaded relief (Imhof, 1982/2007; Jenny and Hurni, 2006), etc.)?
- What parameterization of an image segmentation algorithm can achieve the best performance, while providing a reasonable quality?
- What is the contribution of an image segmentation algorithm to the whole performance of the raster-based method?

The proposed measure is designed for axis-aligned labels. However, cartography deals with more complex shapes of labels to annotate areal (Chapter 6; Pinto and Freeman, 1996; Barrault, 2001), linear (Chapter 8; Edmondson et al., 1996; Wolff et al., 2001), or even point-like (see Figure 2.2) features. These types of designation incorporate locating of letters that are rotated along a curve. Hence, in order to apply the presented measure for scoring a label placement with rotated letters, a modification to the method is required. This modification consists in resampling and weighing of pixels that lie on the border of an arbitrarily rotated rectangle which describes the bounds of a letter. It is worth noting that the contribution of additional computing operations to the performance of the method should be also investigated.

The runtime measurements showed that the developed raster-based method is appropriable for interactive and dynamic labelling (Been et al., 2006; Mote, 2007). It was shown that the method needs ~0.2 s to score label positions. According to Wardlaw (2010), this time can be considered as reasonable for interactive cartography (Roth, 2013). In addition, the method is quite modest in terms of resource usage. Thus, it can be used for labelling maps on mobile devices (Kovanen and Sarjakoski, 2013). Nevertheless, further insights into its practical usage for a real-world application are needed. For the sake of interactivity and resource consumption, different possible scenarios towards performance improvements should be considered. These scenarios should include parallel scoring of candidate label positions on multiple independent CPUs or performing of image segmentation using the computing power of a GPU (Backer et al., 2013). Moreover, a client-server model can be potentially employed. This implicates that some steps involved in the developed measure can be effectively divided between a client and server. For example, image segmentation can be performed on the server side in advance and only then a segmented map is delivered to the client. The client actually accomplishes all the tasks of automated labelling (Section 8.2.1) such as label-position generation, evaluation and selection (Edmondson et al., 1996).

The experimental results showed that the suggested approach is fast. However, it is not clear yet how fast it is in comparison to a vector-based approach, since the tested closed source solution in Maplex (Section 7.4.3) does not allow performing such a comparison. For future work, it is important to conduct a research that involves a comparison of the raster-based approach against an implementation of the vector-based approach using geographic data at different scales. First, such a comparison should investigate the runtimes of both methods. Second, it would be interesting to examine the difference in the quality of label placement between two methods at different scales. However, this kind of study is exceptionally complicated due to the lack of a well-defined vector-based method that can be found in the literature. Moreover, to the best of our knowledge, there is no implementation of the vector-based method that is publicly available.

2.4 An Approach for Labelling Geographic Boundaries

The following subsections introduce the research results that have been given in the paper:

Rylov, M.A., and A.W. Reimer. 2014. "Pairwise Line Labelling of Geographic Boundaries: An Efficient and Practical Algorithm." *Cartographic Perspectives* (submitted)

2.4.1 Results

This part of the research (Chapter 8) introduces an efficient and practical algorithm for producing pairwise labelling of linear features representing the boundaries of demarcated areas (Figure 8.1). As usual, the algorithm consists of two parts which are label-position generation (Section 8.2.4) and evaluation (Section 8.2.5; van Dijk et al., 2002). Label-position evaluation is performed using a quality function which was specially designed for a given task by taken a rich set of cartographic guidelines (Section 8.2.2) into account. The simulated annealing algorithm (Christensen et al., 1995) has been chosen to solve a combinatorial optimization problem (Section 5.3.1), namely, to find a good approximation to the global optimum of the quality function. The experiments have been conducted using a real-world dataset of administrative boundaries[12] taken from the *OpenStreetMap* project. A specially designed console program named OSMBoundariesConverter has been used to pre-process OSM data. This program iterates through a set of polygons by extracting the shapes and the names of administrative boundaries that cor-

---

[12] http://wiki.openstreetmap.org/wiki/Tag:boundary%3Dadministrative

respond to two adjacent regions. In the conducted experiments, two aspects were considered with regard to the performance of label-position generation and to the quality of label placement.

The performance measurements showed that the procedure for generating label positions performs very fast (see Figure 8.11b). According to the measurements, the time to compute one label position using different settings falls into the range [0.0053 s, 0.0076 s]. The experimental results revealed that the time for position generation does not depend on the quality threshold Q_{T}. In turn, Q_{T} has a greater influence on the number of generated label positions (Figure 8.12a). Furthermore, the number of generated labels only slightly depends on the parameterization of the quality function. Also, it was found that the baseline offset has a great influence on the number of generated labels (Section 8.2.5.2). It is also worth noting that the developed algorithm is able to assign label placement in 95% of all possible positions using a certain set of input parameters.

It was observed that the devised quality function (Section 8.2.3) performs its task quite well. Thus, the position-selection process tends to find a cartographically plausible labelling. The function aims to give a higher score value to positions that are placed along more straight segments of the linear feature. Thus, those regions, where a linear feature has a jagged shape, are less attractive for placing labels. Furthermore, it was demonstrated that using a rich set of parameters, an expert cartographer is able to control the quality of placed labels, as well as their number. Note, the number of candidate label positions have a great effect on the runtime of a labelling algorithm (see Section “Computational Results” in Ebner et al., 2003).

2.4.2 Discussion

In general, the presented algorithm showed its ability to perform visually plausible and functional pairwise. However, the algorithm has some limitations and weaknesses (see Figures 8.13 and 8.14). Namely, some labels overlap corresponding geographic features. This can be explained by the absence of checks for overlapping between labels and their linear features. In order to remedy these limitations, a post-processing step is needed. To accomplish this task, the algorithm by Shamos and Hoey (1976) (see also Bentley and Ottmann, 1979) can be utilized. This algorithm reports intersecting points between two sets of line segments. In our case, the first set represents a linear feature itself. The second one corresponds to a set of line segments which form the boundaries of a label (8 line segments per pair of labels). It is worth noting that aforesaid step requires additional computational resources which can lead to some decrease in the performance.

The developed algorithm includes four weights in the quality function (equation (8.2)) and many parameters which have to be configured by a user. Some cartographers can probably question that it might be tedious to tune a half-dozen of input parameters that are needed to be set up only for boundary labelling. In general, the number of configurable parameters can be significantly reduced. More precisely, the distance between repeated labels S can be computed in the runtime by considering both the actual length of the text and some minimum permissible distance. Moreover, the parameters D_{max} and D_{step} (Section 8.2.4.1) for defining the search space can be hard coded by a software developer, but their optimal values should be first empirically defined by a skilled cartographer. This also holds true for the four weights that corresponds to each quality metric (Section 8.2.5.1-8.2.5.4). Furthermore, some predefined sets of weights can be constructed and made available through the user interface (Figure 2.4). Then, the cartographer can choose one of them, which is more appropriate for a certain task and one's preferences. The rest of parameters such as $B_{\mathrm{min}}, B_{\mathrm{max}}$ and Q_{T} may be specified through some default values. Concluding this study outcome, a possibility to adjust the parameters gives a cartographer more flexibility and freedom in designing map lettering. This has been demonstrated in Chapter 5 (Section 5.4). It was shown that both the number of considered metrics and their preferences affect the final labelling.

Chapter 8 introduces an algorithm to label boundaries with a label aligned to a straight line as it is depicted in Figure 8.1. Cartography, however, has several guidelines for labelling these geographic features. A more common technique is to annotate boundaries using curved labels. Unfortunately, the literature does not contain meaningful guidelines for this task. Nonetheless, one can definitely find many examples of this technique in printed maps and atlases (see Perry-Castañeda Library Map Collection[13]; for instance, see Figure 8.2). This implies the requirement of another algorithm to cope with curved labels. A new algorithm should include its own procedure for generating candidate positions. Moreover, as the shape of a label is curved, a new quality function is needed in this case. It is obvious that this function should incorporate another set of cartographic rules. In conclusion, the results of the presented work can significantly facilitate the development of a new labelling algorithm and help to perform an analysis of its cartographic quality and performance.

2.5 Implementation

This section gives a short overview of the implementation of the presented algorithms.

2.5.1 Integration of the Methods into a Framework

All proposed models and methods are implemented within a framework which is called *MapSurfer.NET* (2014). This framework is intended for visualizing geospatial data on a

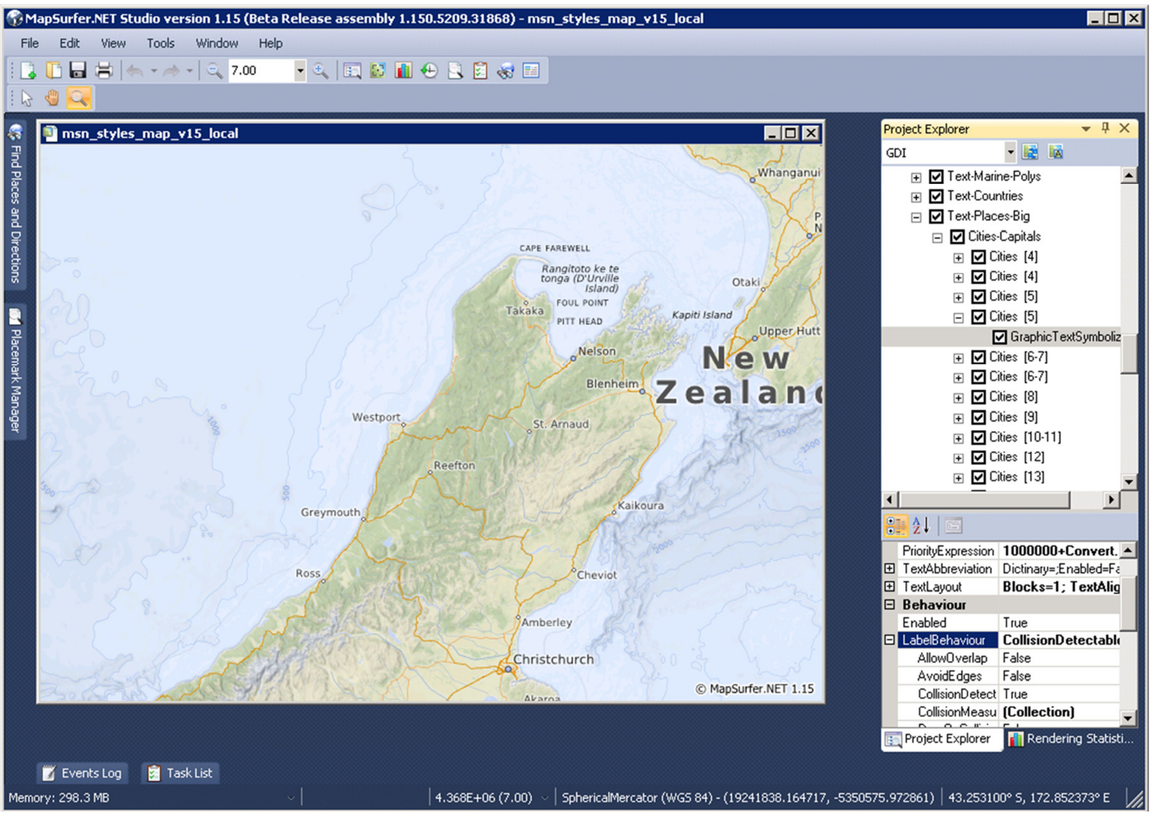
---

[13] http://www.lib.utexas.edu/maps

Figure 2.3. The diagram of label placement classes which are implemented in MapSurfer.NET. The classes *PointPlacement*, *HorizontalOutsidePolygonPlacement*, *DoubleSidedLinePlacement* correspond to the procedures for label-position generation in Chapters 5, 6, 8 respectively.

desktop or for publishing them to the web. It is written in C# programming language. The framework is partially based on some Open Geospatial Consortium's (OGC) standards such as Styled Layer Descriptor (SLD; OGC, 2007), Symbology Encoding (SE; OGC, 2006), Standard for Geographic Information (OGC, 2011), Web Map Service (WMS; OGC, 2006) and Web Map Tile Service (WMTS; OGC, 2010). These standards were designed for symbolization and for portraying geographic information to distribute thematic mapping by means of web services (Sykora et al., 2007).

The parameterization of the developed algorithms is done through an extension for Symbology Encoding specification. Symbology Encoding is intended for controlling the appearance of a resulting map. An extension to *LabelPlacement* element (see Section 11.4.4 in OGC, 2006), which is similar to the existing elements *PointPlacement* and *LinePlacement*, has been proposed for each developed algorithm (see for example Rylov and Zipf, 2012). As it can be seen from Figure 2.3, the labelling capabilities of MapSurfer.NET are not limited to the presented models and methods. The capabilities cover almost all classes and subclasses of label placement tasks (e.g., points, lines, areas) that are



PriorityExpression	**1000000+Convert.**
TextAbbreviation	Dictinary=;Enabled=Fa
TextLayout	**Blocks=1; TextAlig**
Behaviour	
Enabled	True
LabelBehaviour	**CollisionDetectabl**
AllowOverlap	False
AvoidEdges	False
CollisionDetect	True
CollisionMeasu	**(Collection)**

Figure 2.4. A user interface of MapSurfer.NET Studio application for creating and editing map styles, including a flexible configuration of the labelling settings

established both in automated and in manual labelling. The diagram in Figure 2.3 shows a set of parameters to control label-position generation. What about the parameterization of the quality metrics (for example, Sections 5.3.4.1 - 5.3.4.6) and the weights of a general quality measure (see equation (1.1))? In the given implementation, one can configure both of them through the user interface (Figure 2.5) of a specially designed desktop application (see Figure 2.4). Note that SE standard does not have a specification to configure the mentioned parameters of a labelling algorithm. Furthermore, there is a great diversity of formats (OGC's Symbology Encoding) and map styling languages (MapServer's Mapscript[14], CartoCSS[15], MapCSS[16], etc.) in the existing geographic information systems, libraries and servers such as ArcMap, MapInfo, QGIS, Mapnik, TileMill, MapServer, GeoServer, etc. However, only a few of them partially support parameters that are specific for more advanced labelling.

---

[14] http://mapserver.org/el/mapscript/index.html

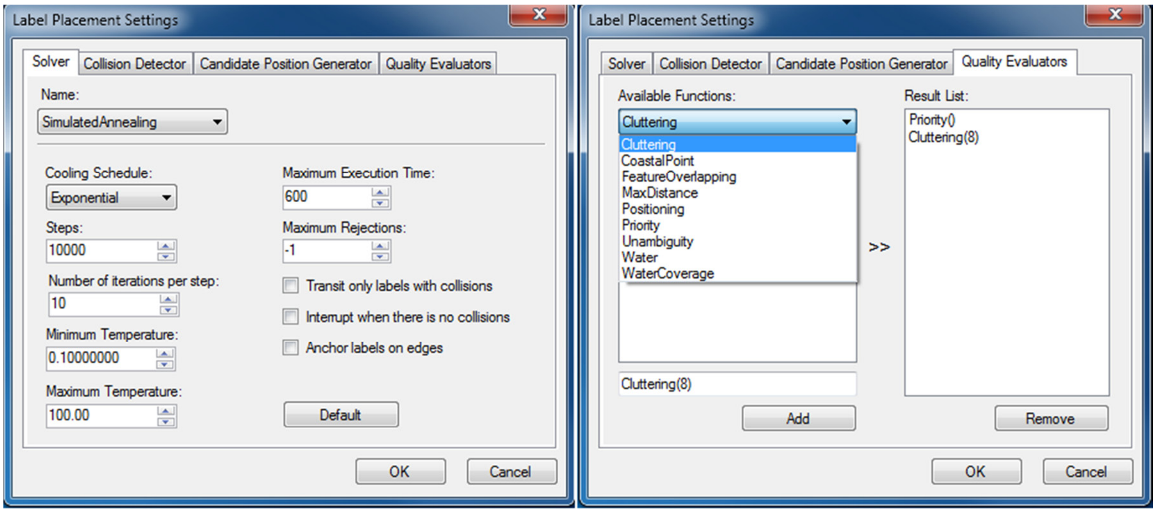
[15] https://www.mapbox.com/carto/api/2.3.0

[16] http://www.mapcss.org

It is highly important to mention that the developed framework is not limited only to solve map labelling problem. This framework is a handy tool which helps to automate a wide range of cartographic problems and tasks specified in the works by Imhof (1972), Keates (1973), Robinson et al. (1995). These problems comprise, but are not limited to:

- feature and toponym selection (quantity and type) in respect to an intended map-use task;
- typeface configuration (e.g., font choice, font form (spacing, colouring, italics, etc.), font size);
- feature representation (e.g., colour choice, stroke width and cap style of lines; polygon fill, etc.);
- geometric placement of labels;
- cartographic hill shading representation;
- feature transformation (e.g., simplification, clipping, etc.);
- feature reprojection (geometry and raster);
- adjustment (alterations of settings and map regeneration).

Therefore, according to the formalized criteria given by Forrest (1999), the framework can indeed be considered as a cartographic expert system. The framework has many features and is able to provide a map product (e.g., geological (Lisle, 2003), cadastral, high way or city street maps, etc.) of high cartographic quality, but it is still in a prototype phase. To become a tool that can be extensively used by diverse users, a thorough documentation of its capabilities, features and settings has to be written. Further, the developed framework includes both a Software Development Kit (SDK) and a ready-to-use



Name: SimulatedAnnealing

Cooling Schedule: Exponential

Steps: 10000

Number of iterations per step: 10

Minimum Temperature: 0.10000000

Maximum Temperature: 100.00

Maximum Execution Time: 600

Maximum Rejections: -1

☐ Transit only labels with collisions

☐ Interupt when there is no collisions

☐ Anchor labels on edges

Available Functions: Cluttering

Cluttering(8)

Result List:
Priority()
Cluttering(8)

(a) (b)

Figure 2.5. The elements of the user interface in MapSurfer.NET Studio to parameterize label placement settings: a) Solving algorithm; b) Quality metrics.

application called *MapSurfer.NET Studio* (Figure 2.4). This application can be used by GIS specialists, experienced cartographers or even by users unfamiliar with cartographic design principles. Moreover, one can use the framework for preparing a cartographic product and then publishing it to the web or for embedding its functionality in one's own solution. MapSurfer.NET has a great potential to assist in cartographic education to improve the experience of cartographers in map-making with the aid of a computer. Certainly, one can also use the framework for the purpose of conducting further research in the field of label placement. Since, any part of a labelling algorithm (label-position generation, evaluation, selection) can be easily substituted for a custom component using a flexible plugin system of the framework.

It should be mentioned that the most part of the maps used in this dissertation have been styled and rendered using MapSurfer.NET framework. Furthermore, the framework has been entirely developed by the author of the dissertation.

2.5.2 A Web Service for Map Visualization

Many diverse experiments on different and geographically scattered mapped areas have been conducted within the presented research. The conducted experiments were not only restricted to the results given in each chapter. The developed methods and models have been tested and assessed on different regions all over the world. The results of the conducted research further encouraged the decision to develop a web map service that could provide an online map for the entire globe with labelling of a superior quality. A brief description of this web service is given in the following paragraphs.

The developed service (see Figure 2.6), named OpenMapSurfer[17], consists of the following components:

- Data sources
  - OpenStreetMap data
  - ASTER GDEM and SRTM digital elevation models
  - GlobCover – a land cover map
  - GeoNames – a database of geographical names
- Software libraries and packages
  - GDAL library
  - Osm2pgsql converter
  - PostgreSQL server
  - .NET Framework 4.5
  - MapSurfer.NET framework
  - Internet Information Services (IIS)

---

[17] http://openmapsurfer.uni-hd.de/

- Windows Web Server 2008

In more details, a command-line based program called *osm2pgsql*[18] was used. This program converted and stored OpenStreetMap data to a PostgreSQL database with PostGIS spatial extension. To initially fill a spatial database with geographic features, a complete planet dump file was used. This file was downloaded from Planet OSM[19] page. The land areas for the whole world have been taken from OpenStreetMap Data[20] website. Additionally, ASTER GDEM and SRTM digital elevation models were downloaded from the web and stored on the file system.

MapSurfer.NET Studio (Figure 2.4) was used to design map styles. Note that MapSurfer.NET framework supports many geographic data source types (for example, ESRI Shape, GPX, OpenStreetMap's *.xml or *.pbf, GeoJSON, MySQL spatial extension, CartoDB, etc.). The visual portrayal of geographic features has been done for 20 fixed scales which are common in web maps (OGC, 2010, p. 10; García et al., 2012). Next, as it was already mentioned, the framework has a partial implementation of OGC's Web Map Tile Service (2010) specification. Thus, MapSurfer.NET was used to set up a web

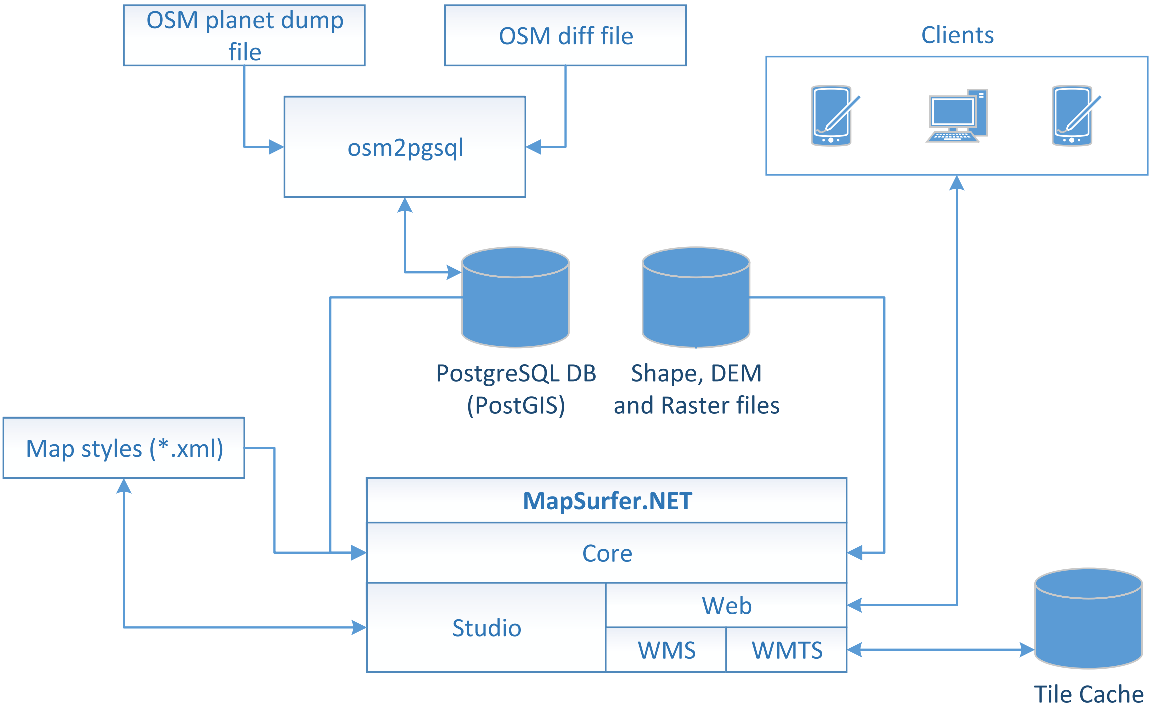


Figure 2.6. The workflow of the developed web map service.

---

[18] http://wiki.openstreetmap.org/wiki/Osm2pgsql

[19] http://planet.osm.org/

[20] http://openstreetmapdata.com/data/land-polygons

service with the help of Internet Information Services (IIS, a web server created by Microsoft). A general workflow of the web map tile service is illustrated in Figure 2.6. Web clients can request geo-referenced images through a HTTP interface. In our case, the service receives the coordinates of an image, known as a tile, and sends a response containing an image, which is a part of a map at a certain scale. As it can be seen from Figure 2.6, an approach of tile caching (García et al., 2012, pp. 28-35) was used in the implementation. This approach was designed to support repetitive requests in order to save resources and significantly increase the performance of a web map service (Loechel and Schmid, 2013).

The web map tile service on OpenMapSurfer is running since January 2012. At the moment the service handles about 5.5 millions of requests every day. Its database of geographic features is updated every hour using osm2pgsql converter which applies changes from so-called *OSM Change-Files* (OSC), also known as *Diff-Files*, which contain latest edits made by contributors. The tile cache is kept and maintained up-to-date by re-rendering those tiles in which some changes have occurred. Both update processes are performed in the background. This approach let web clients continue working without noticing any significant delays.

2.5.3 Results

As a result, the underlying idea and the objective of the developed service (Section 2.5.2) were to demonstrate to a broad audience the results of conducted research. This service shows that proper interpretation and formalization of cartographic design principles for label placement helps to produce a high-quality online map.

The main map which is available as "OSM Roads" layer on OpenMapSurfer service has the following benefits in comparison to other popular free and proprietary online maps provided by different map providers such as OpenStreetMap, Google Maps, Microsoft Bing Maps, HERE Maps, Mapbox, etc.

- Greater number of labels for point features at small scales, since the proposed methods are able to produce a map with a higher density of labels (see Figure 2.8).
- Fewer labels that have an ambiguous relationship with the neighbouring labels and their associated symbols. It is achieved with the help of the specially designed metric given in Section 5.3.4.4.
- The labelling is much more readable and more legible due to use of the background information (Chapter 7 or Section 5.3.4.3). For example, road network and water bodies are less obscured or concealed by the labels. Since, the labels for coastal places are placed in the water while other labels are located in the regions free from background features (see Figure 2.8).

- More advanced algorithms give the possibility to annotate those features which are normally omitted or poorly labelled on the competitive maps (see Figure 2.9), especially at small and medium scales.

Map Compare[21] tool gives a great opportunity to the reader to perform a visual comparison between diverse online maps including “OSM Roads”. The labelling on this map has been produced by incorporating all proposed methods into a comprehensive labelling solution as one whole.

The usage of the comprehensive methods in web mapping with a tile-based approach (García et al., 2012) raises a problem of label placement consistency on the boundaries of map tiles (Figure 2.7). In a tile-based approach, a map is usually rendered and labelled by pieces which are defined by the boundaries of the specific tiles (or metatiles). Therefore, each map region is processed independently, which ultimately means that a labelling algorithm has no information about label placements in the neighboring map regions. This problem is not new and is known for some time[22]. Moreover, almost all existing toolkits for publishing maps to the web (ArcIMS, GeoServer, MapServer, Mapnik, etc.) have similar problems. However, it was noticed that the problem of cut off labels is comparably greater when one applies the proposed models. Many more labels were observed than in the case of using a straightforward labelling by exploiting a greedy algorithm (Section 3.2 in Christensen et al., 1995). Concluding, some further research regarding an application of sophisticated labelling models for tile-based maps needs to be conducted.

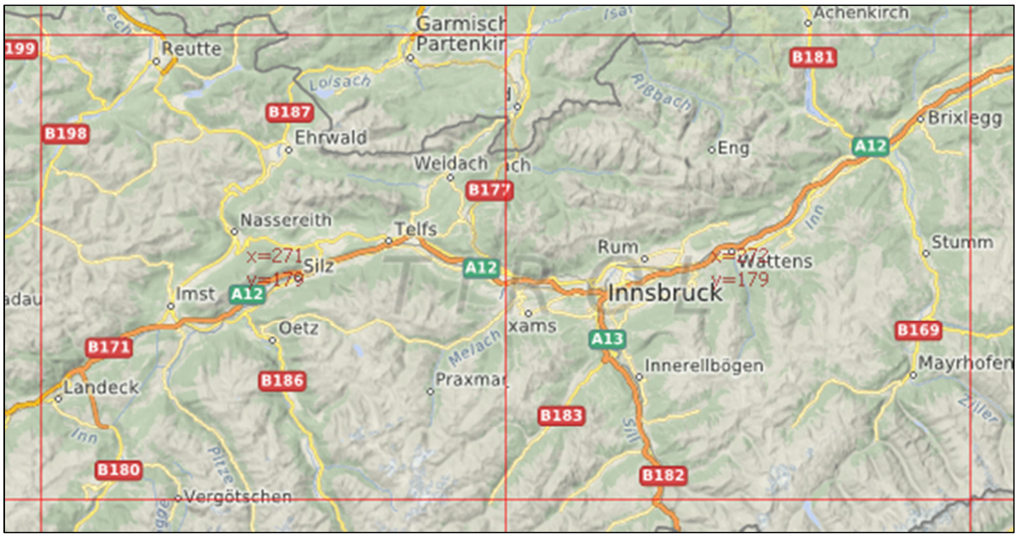


Figure 2.7. An example for cut off labels at tile boundaries. A generated map labelled by using a labelling model presented in Chapter 5.

---

[21] http://mc.bbbike.org

[22] http://resources.arcgis.com/en/help/main/10.1/index.html#//015400000358000000

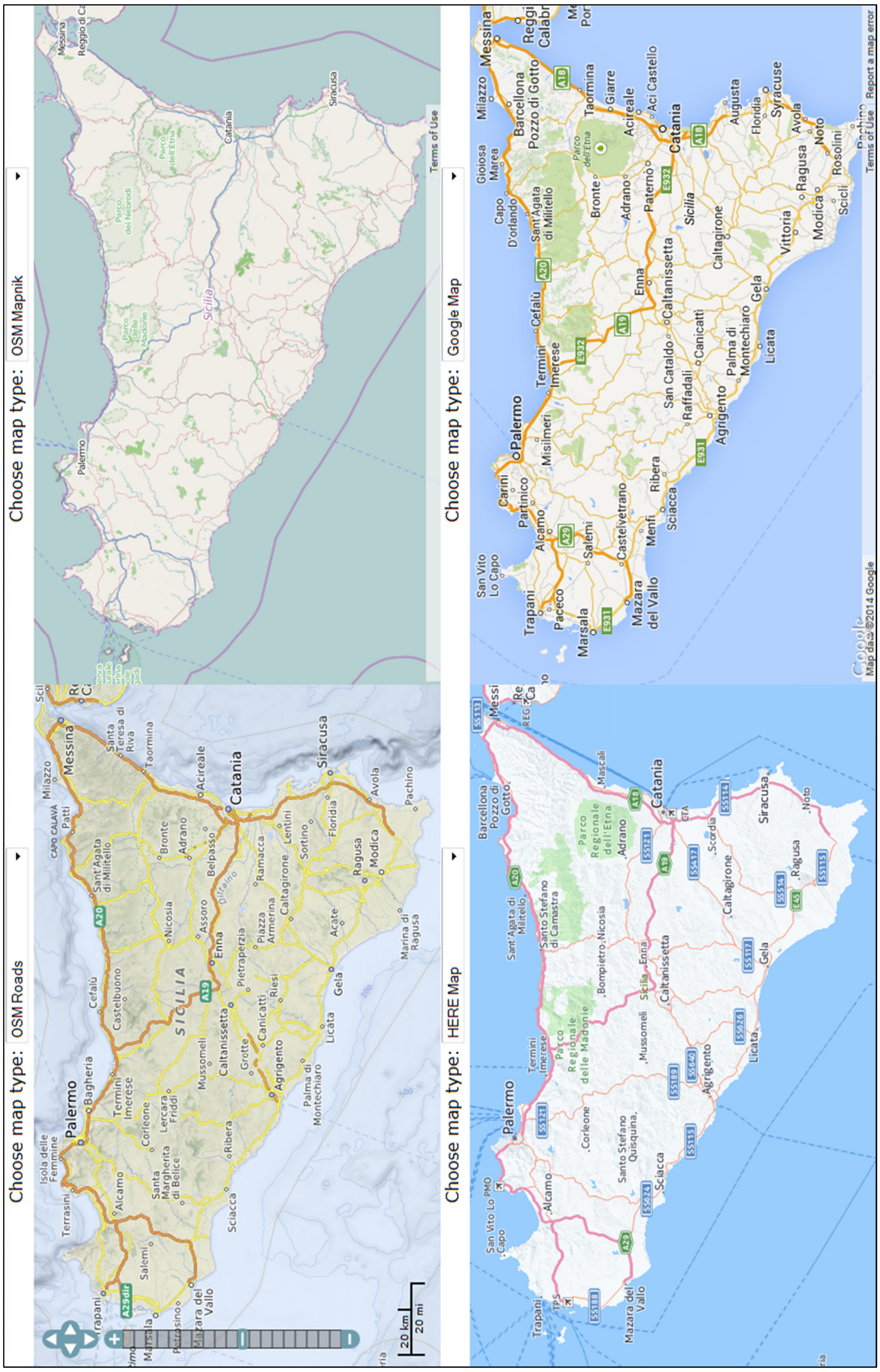


Figure 2.8. The island of Sicily depicted on different online maps. Number of labelled settlements: OSM Roads – 56, OSM Mapnik – 5, HERE Map (Nokia) – 28, Google Map – 54. Note that the layers OSM Roads and OSM Mapnik use the same dataset.

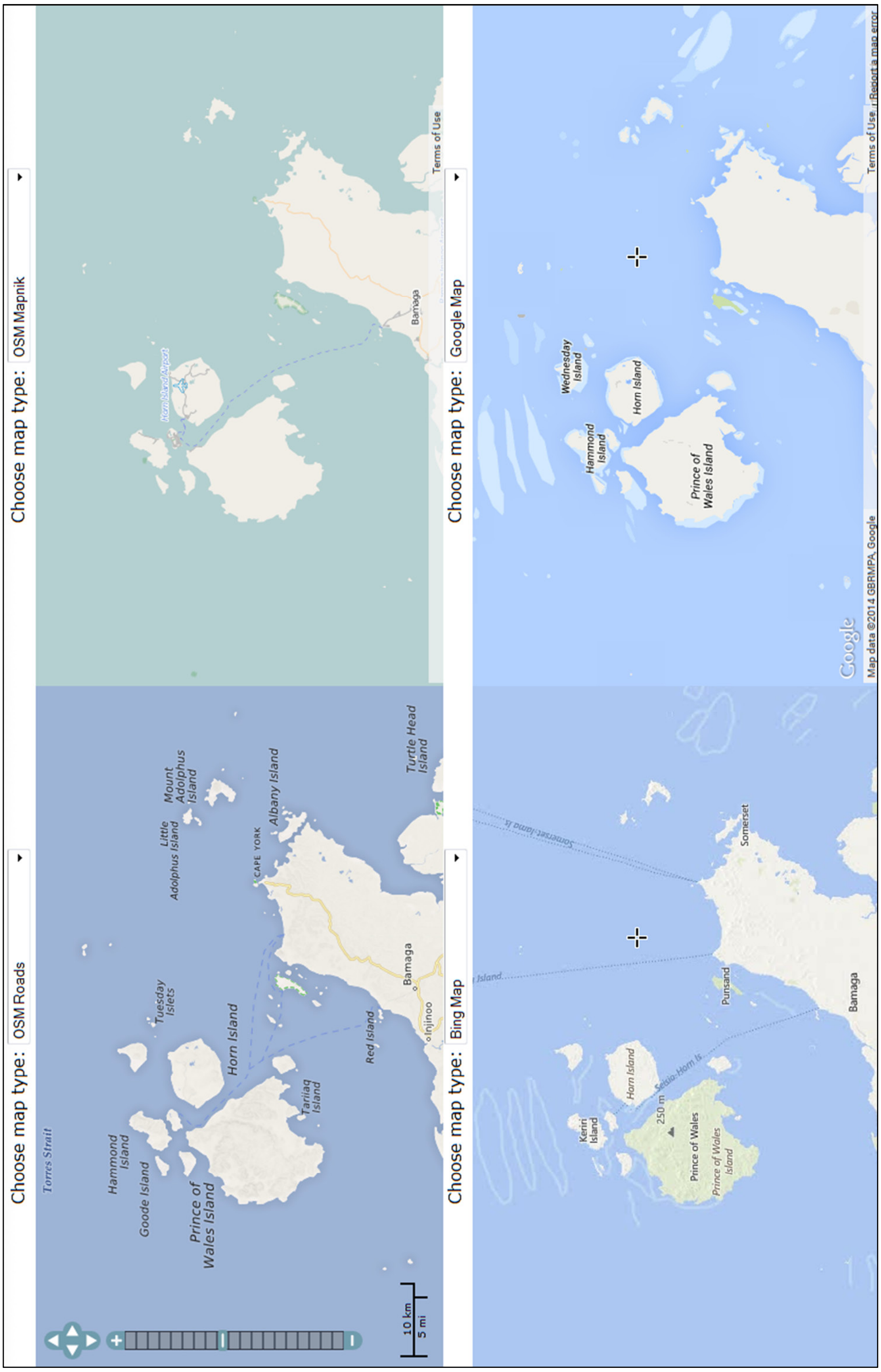


Figure 2.9. A comparison of labelling on different online maps. The map depicts a part of the Torres Strait Islands.

3 Conclusions

In spite of many preceding research attempts to fully automate the task of map labelling, this specific problem is still vital and challenging. Being a many-sided, the problem of map lettering is split into subtasks and consists of various building blocks. Each of these blocks corresponds to one or several design techniques used in manual cartography.

One of the main objectives of this dissertation was to understand why existing algorithms underperform in terms of good label placement. This dissertation elaborates some missing blocks and makes an attempt to advance automated label placement towards a quality of name placement on hand-drawn maps. To achieve this objective, a number of newly developed models and methods were introduced. Each presented method uses accumulated knowledge in the related field, as well as algorithmic innovations in mathematical optimization techniques, computational geometry, computer graphics and new technologies in Web GIS (Fu and Sun, 2010) to present the resulting maps.

Point-feature label placement (PFLP) is an essential part of any map presented at small scale. A thorough review of the existing methods for solving PFLP (Sections 5.2 and 5.3.2) has revealed that the problem of PFLP was solved incompletely. The developed model (Chapter 5) introduces a first attempt to simultaneously consider all cartographic guidelines used in manual lettering of point features. The model presents a novel metric that measures ambiguous relationships between symbols and their names. Note that this cartographic guideline not only has a primary importance in label placement but also occurs in various disciplines such as graph (Battista et al., 1994; Kakoulis and Tollis, 2003) and schematic diagram drawing (Nöllenburg and Wolff, 2011; Fink et al., 2012), as well as GIS (Freeman, 1991) and 3D modelling (Götzelmann et al., 2005; Lehmann and Döllner, 2012). However, the mentioned guideline has been completely neglected in the preceding research (Table 5.1). Next, the experimental results demonstrated the contribution of each presented metric in terms of the overall performance of the labelling algorithm (Table 5.3). These results can play an important role in understanding the complexity of computations. Therefore, this should facilitate further research towards the optimization of internal parts of a labelling algorithm that utilizes advanced quality metrics.

The developed multi-criteria model comprises a simple version of a raster-based approach that takes label-feature overlap into account (Section 5.3.4.3). This approach demonstrated good results in the conducted experiments and impelled us to improve the raster-based approach in respect to considering more cartographic guidelines. As a result, a novel generic quality evaluation model (Chapter 7; Rylov and Reimer, 2014c) was introduced. This model allows quantifying map feature overlap and visual clutter between labels and map background. The results of experiments exemplified that the proposed extended version of the raster-based model is not only highly efficient but also comparable in terms of quality to labelling using a vector-based approach (Strijk and van Kreveld, 2002). Furthermore, the presented approach opens new fields of potential application such as lettering maps with relief shading (Imhof, 1982/2007; Jenny and Hurni, 2006) and bathymetry, or even for annotating 3D models (Lehmann and Döllner, 2012). Note that the introduced model has been designed in a form which makes it possible to use it as a component for a general quality evaluation function described by van Dijk et al. (2002).

Two new methods for labelling areal and linear features were developed in the course of this work. The necessity of their development was conditioned by the absence of appropriate methods in literature. The first method given in Chapter 6 presents an algorithm which is able to label areas outside their boundaries. Besides its high performance, the algorithm is able to produce visually appealing and, which is more important, unambiguous label placement. The conducted comparative study confirmed that this algorithm outperforms both tested commercial packages Maplex and Label-EZ in the quality of name placement (Section 6.4.4). The second method is described in Chapter 8. This algorithm is aimed to perform pairwise line labelling of geographic boundaries. In terms of practical applicability, the presented method proved itself as a convenient tool for a human cartographer to accomplish a certain task. Note that none of the freely available open source toolkits for map rendering has either equivalent of the two mentioned labelling methods. Hence, those toolkits could benefit from adding the proposed algorithms to their set of labelling features.

Concluding, the most significant findings of this dissertation can be summarized as follows:

- The overall work revealed that automated methods consider cartographic knowledge rather poorly which leads to inaccurate and amateurish map lettering. Furthermore, it has been detected that not every design technique used in manual text placement has its automated counterpart.
- The developed multi-criteria model highlights that the profound consideration of cartographic principles and their careful formalization through a set of quality metrics can help to achieve a new qualitative level of label placement produced by means of computers. The development of more advanced metrics is very promising and highly demanded.
- Labelling algorithms should place higher emphasis on the importance of unambiguous relationship between neighbouring labels and the corresponding map features. The results showed that the presence of an appropriate measure enhances the readability of a map considerably. This is especially true for maps with a high density of labels (Figure 1.2).

- The proposed raster-based method for considering basemap detail (Chapter 7) showed its applicability in practice. It was proved that this method highly increases both readability and legibility of a map.
- The comparative study in Section 7.4.3 highlighted that a raster-based method is competitive to a vector-based approach in terms of the quality of the resulting label placement. In addition, it has further potential to be utilized in various use cases.
- A novel algorithm for annotating areal features outside their boundaries was presented (Chapter 6). The algorithm includes a procedure for a fast and efficient generation of potential label positions as well as the quality function for their scoring.
- A novel practical algorithm for pairwise labelling of geographic boundaries was introduced (Chapter 8). The proposed algorithm gives the opportunity to label a linear feature in a very specific way. Note that until now none of the previously published methods is able to accomplish this cartographic task.
- The experiments highlighted the necessity of exploiting computational power of modern GPUs to perform different labelling subtasks which involve image segmentation (Backer et al., 2013) or combinatorial optimization (Cavuoti et al., 2013; Ferreiro et al., 2013).
- A visual comparison of labelling on different online maps has been performed (Section 2.5.3). The comparison showed that almost all frameworks which are used for publishing maps to the web are rather modest in accomplishing cartographic labelling properly. The labels are comparably sparse; moreover, they neither consider neighbouring labels and the shape of the tagged features nor specific background information. As a result, many labels tend to be poorly positioned. In turn, the labelling derived from the presented methods is more advanced, since they incorporate more cartographic principles.
- An application of a sophisticated labelling algorithm in web maps requires a technical solution to get rid of cut off labels at the boundaries of tiles in the case of the tile-based representation of a map.

The conducted research has uncovered many aspects of manual lettering that have not been automated yet. The results of the work refute a widely diffused opinion among GIS specialists who believe that the problem of automated label placement was solved many years ago. No, it is not. Indisputably, the algorithms became more sophisticated and matured, but they are still far away from producing labelling which is fully comparable to a good one made by a skilled map-maker. This is especially true for maps with a high graphical density. Albeit long-standing research in the field of automated labelling, the words by Wood which state that “no one algorithm seems to be capable of recognizing the many considerations that a skilled human cartographer is capable of making in lettering a map. Thus, while automated type placement has improved dramatically, and rapidly, it is still not ‘there’ yet, especially for complex small-scale thematic maps” seem still to be valid nowadays. There are many open or even unimpaired questions that require further challenging work.

The results of this research can help to understand why and especially where we have failed in automation of label placement. The novel methods may bridge the gap between currently used algorithms on the market and manual map lettering of professional cartographers. The findings of this dissertation, identified throughout previous chapters, may encourage and stimulate further research in this attractive field.

4 Future Work

The following chapters in Part II (Publications) already provide extensive discussions regarding possible extensions, uses cases, improvements of the presented models and methods. A summary of open research questions, that need to be investigated in future work, is given within the following paragraphs.

An unambiguous relationship between neighboring labels and features has an utmost importance in cartography. Therefore, quantifying this relationship (Sections 5.3.4.4 and 5.3.4.5) is absolutely necessary. However, the computation of the corresponding metrics is computationally very expensive (Section 5.4), especially for map regions with a high density of point features. The implementation of our measure is rather straightforward. This was confirmed in the experiments. Hence, a more sophisticated, more effective and faster algorithm to tackle this problem is needed. Next, a first attempt to measure the degree of ambiguity between labels was undertaken. Table 5.1 entirely confirms this statement. The developed model only considers the case of “point-point” relationship. Thus, the development of appropriate metrics for other possible types of relationship such as “point-line”, “point-area”, “line-line”, “line-area”, “area-area” is a topic for subsequent research.

A novel algorithm for annotating areal features externally (Chapter 6) was introduced. It is worth noting that this is the first published work to automate this corresponding cartographic task. Despite this algorithm outperforms similar solutions of commercial packages (Section 6.4.4), some further improvements could be suggested. This includes the elaboration of new proximity measures for the quality function and the comparison of various proximity measures in terms of their ability to reflect the degree of spatial relationship between a label and the feature it tags.

The proposed raster-based approach (Chapter 7) for considering overlap and concealment of non-textual background features by labels proved to be rather efficient in the experiments. However, some questions remain open and require further research. First, a modification of the model for supporting curved labels is still needed. Second, the influence of an image segmentation algorithm and its parameterization on the resulting labelling and on the whole performance of the labelling algorithm is not fully comprehensible. Third, it would be very interesting to thoroughly compare the performance of the raster-based approach with a vector based approach (Strijk and van Kreveld, 2002) on maps with different densities of background features. Furthermore, a more detailed comparison of two approaches in face-to-face manner would help to comprehend their merits and flaws.

For unknown reasons, previous research completely omitted pairwise labelling for linear features. The accomplished attempt to make a first step in this direction has proved itself as a practical and efficient solution (Chapter 8). But, the presented solution has particular deficiencies such as the placements of labels which intersect boundaries when the shape of a given polyline is complex and sinuous. To address this problem, it was suggested to exploit the algorithm by Shamos and Hoey (1976). Moreover, some details of how this could be done are provided. But, this is just a proposal to handle the problem which undoubtedly provides an insight for new research questions. One of these questions, for example, is to see its impact on the performance of the pairwise labelling algorithm. Furthermore, our developed method solves the problem of pairwise labelling only partially, i.e. the labels are placed along an imaginary straight line. Therefore, an algorithm that is able to place coupled labels in a curved fashion is a formidable challenge for future research.

In summary, the conducted experiments on the global datasets have emerged the necessity of a faster algorithm for solving a combinatorial optimization problem. The performance of a labelling algorithm, for instance in the use case of OpenMapSurfer, plays an important role in map rendering process as labelling has to be applied for a map of the globe. Hence, the production of such a map becomes time-consuming. It was observed in the experiments that the labelling process uses the majority of time in the map rendering, in some cases even up to 90%. This is especially true for populous regions with a high density of label candidates. Furthermore, the labelling of such regions, on a web map with the tile-based representation, raised a problem of cut off labels at the boundaries of adjacent tiles (see Section 2.5.3). Therefore, some further research towards a technical solution to cope with this problem is highly required.

References

Agarwal, P.K., M. van Kreveld and S. Suri. 1998. “Label placement by maximum independent set in tractangles.” *Computational Geometry* 11(3-4):209-218. doi:10.1016/S0925-7721(98)00028-5

Ahn, J., and H. Freeman 1984. “A Program for Automatic Name Placement.” *Cartographica* 21(2-3): 101-109. doi: 10.3138/0646-Q262-6636-3681

ASTER Validation Team. 2011. ASTER Global Digital Elevation Model Version 2 – Summary of Validation Results. Published by the ASTER GDEM Validation Team: METI, NASA and USGS. August 2011, 27 pages. Available online at: http://www.jspacesystems.or.jp/ersdac/GDEM/ver2Validation/Summary_GDEM2_validation_report_final.pdf. http://reverb.echo.nasa.gov.

Backer, M., J. Tünnermann, and B. Mertsching. 2013. “Parallel k-Means Image Segmentation Using Sort, Scan and Connected Components on a GPU.” In *Facing the Multicore-Challenge III*, 108-120. Berlin-Heidelberg: Springer. doi: 10.1007/978-3-642-35893-7_10

Barrault, M. 2001. “A Methodology for Placement and Evaluation of Area Map Labels.” *Computers, Environment and Urban Systems* 25(1): 33-52. doi: 10.1016/S0198-9715(00)00039-9

Barrault, M., and F. Lecordix. 1995. “An Automated System for Linear Feature Name Placement which Complies with Cartographic Quality Criteria.” In *Proc. AutoCarto 12*: 321-330. ACSM/ASPRS, Bethesda.

Basoglu, U. 1982. “A new approach to automated name placement.” In *Proc. AutoCarto 5*: 103-112.

Battista, G.D., P. Eades, R. Tamassia, and I.G. Tollis. 1994. “Algorithms for Drawing Graphs: An Annotated Bibliography.” Computational Geometry: Theory and Applications 4(5):235-282. doi: 10.1016/0925-7721(94)00014-X

Been, K., E. Daiches, and C. Yap. 2006. “Dynamic Map Labeling.” *IEEE Transactions on Visualization and Computer Graphics* 12(5): 773-780. doi: 10.1109/TVCG.2006.136

Bentley, J.L. and T.A. Ottmann. 1979. “Algorithms for Reporting and Counting Geometric Intersections.” *IEEE Transactions on Computers* C–28(9): 643-647. doi: 10.1109/TC.1979.1675432

Bo, Q. 2010. "Recursive polygon offset computing for rapid prototyping applications based on Voronoi diagrams." *The International Journal of Advanced Manufacturing Technology* 49(9-12): 1019-1028. doi: 10.1007/s00170-009-2454-4

Brewer, C.A. 2005. Designing Better Maps: A Guide for GIS Users, ESRI Press, Redlands CA.

Brown, B. 1976. "Effect of Background Constraint on Visual Search Times." *Ergonomics* 19(4):441-449. doi: 10.1080/00140137608931556

Carstensen, L.W. 1987. "A Comparison of Simple Mathematical Approaches to the Placement of Spot Symbols." *Cartographica* 24(3): 46-63. doi: 10.3138/GH04-24H7-63T8-167V

Castner, H.W. and R.J. Eastman. 1985. "Eye-Movement Parameters and Perceived Map Complexity—II." *American Cartographer* 12(1): 29-40. doi: 10.1559/152304085783914712

Cavuoti, S., M. Garofalo, M. Brescia, A. Pescape', G. Longo and G. Ventre. 2013. "Genetic Algorithm Modeling with GPU Parallel Computing Technology." *Neural Nets and Surroundings Smart Innovation, Systems and Technologies* 19: 29-39. doi: 10.1007/978-3-642-35467-0_4

Chazelle, B. 1999. "The computational geometry impact task force report." In *Advances in Discrete and Computational Geometry.* ed by B. Chazelle, J. E. Goodman, and R. Pollack, Vol. 223, 407-463. American Mathematical Society, Providence.

Chen, X. and S. McMains. 2005. "Polygon Offsetting by Computing Winding Numbers." In *ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2005),* September 24-28, Long Beach, California, USA.

Chirié, F. 2000. "Automated Name Placement With High Cartographic Quality: City Street Maps." *Cartography and Geographic Information Science* 27(2): 101-110. doi: 10.1559/152304000783547902

Christensen, J., J. Marks, and S.M. Shieber. 1995. "An Empirical Study of Algorithms for Point-Feature Label Placement." *ACM Transactions on Graphics* 14(3): 203-232. doi: 10.1145/212332.212334

Cromley, R.G. 1985. "An LP Relaxation Procedure for Annotating Point Features Using Interactive Graphics." In *Proceedings of the AutoCarto 7,* 127-132.

de Berg, M., O. Cheong, M. van Kreveld, and M. Overmars. 2008. *Computational geometry: Algorithms and Applications.* 3rd ed. Berlin-Heidelberg: Springer.

Dörschlag, D., I. Petzold, and L. Plümer. 2003. "Placing objects automatically in areas of maps." In *Proc. The 23rd Internat. Cartographic Conf. (ICC'03)*: 269-275.

Ebner, D., G.W. Klau, and R. Weiskircher. 2003. "Force-Based Label Number Maximization." Technical Report TR-186-1-03-02, Vienna University of Technology.

Edmondson, S., J. Christensen, J. Marks, and S. Shieber. 1996. "A General Cartographic Labeling Algorithm." *Cartographica* 33(4): 13-23. doi: 10.3138/U3N2-6363-130N-H870

Ertz, O., M. Laurent, D. Rappo, A. Sae-Tang, and E. Taillard. 2009. "PAL : A cartographic labelling library", PositionIT, EE Publishers (Pty) Ltd, Gauteng, South Africa. pp. 56-61.

ESRI. 2009. "Maplex for ArcGIS. An ESRI White Paper". Available at http://www.esri.com/library/whitepapers/pdfs/maplex-for-arcgis.pdf.

Evermap. 2014. EverName – Cartographic Label Placement Software for MapInfo Professional. Available at http://www.evermap.com/evername.asp.

Ferreiro, A.M., J.A. García, J.G. López-Salas and C. Vázquez. 2013. "An efficient implementation of parallel simulated annealing algorithm in GPUs." *Journal of Global Optimization* 57(3): 863-890. doi: 10.1007/s10898-012-9979-z

Fink, M., J-H., Haunert, A. Schulz, J. Spoerhase, and A. Wolff. 2012. "Algorithms for Labeling Focus Regions." *IEEE Transactions on Visualization and Computer Graphics* 18(12): 2583-2592. 10.1109/TVCG.2012.193

Formann, M., and F. Wagner. 1991. "A Packing Problem with Applications to Lettering of Maps." In *Proceedings of the 7th Annual Symposium on Computational Geometry*, 281-288.

Forrest, D. 1999. "Developing Rules for Map Design: A Functional Specification for a Cartographic-Design Expert System." *Cartographica* 36(3): 31-52. doi: 10.3138/9505-7822-0066-70W5

Freeman, H., and J. Ahn. 1987. "On the Problem of Placing Names in a Geographic Map." *International Journal of Pattern Recognition and Artificial Intelligence* 1(1): 121-140. doi: 10.1142/S0218001487000096

Freeman, H. 1991. "Computer name placement", In *Geographical Information Systems: Principles and Applications*. ed. D.J. Maguire, M.F. Goodchild, and D.W. Rhind, 445-456. Londong: Longman.

Fu, P., and J. Sun. 2010. Web GIS: Principles and Applications. Redlands: ESRI Press.

García, R., J.P. de Castro, E. Verdú, M.J. Verdú, and L.M. Regueras. 2012. "Web Map Tile Services for Spatial Data Infrastructures: Management and Optimization." In *Cartography - A Tool for Spatial Analysis.* ed. C. Bateira. 25-48. doi: 10.5772/46129

Gartner, G., D.A. Bennett and T. Morita. 2007. "Towards Ubiquitous Cartography." *Cartography and Geographic Information Science* 34(4): 247-257. doi: 10.1559/152304007782382963

Gervautz, M., and W. Purgathofer. 1988. "A Simple Method for Color Quantization: Octree Quantization." *New Trends in Computer Graphics*, ed. N. Magnenat-Thalmann and D. Thalmann. Springer Berlin Heidelberg. 219-231. doi: 10.1007/978-3-642-83492-9_20

Glover, F. 1989. "Tabu Search-Part I." *ORSA Journal on Computing* 1(3): 190-206. doi:10.1287/ijoc.1.3.190

Goodchild, M.F. 2007. "Citizens as Sensors: The World of Volunteered Geography." *GeoJournal* 69: 211-221. doi: 10.1002/9780470979587.ch48

Götzelmann, T., K. Ali, K. Hartmann, and T. Strothotte. 2005. "Form Follows Function: Aesthetic Interactive Labels." In *Computational Aesthetics in Graphics, Visualization and Imaging.* 193-200. doi: 10.2312/COMPAESTH/COMPAESTH05/193-200

Haklay, M., and P. Weber. 2008. "OpenStreetMap – User Generated Street Map", IEEE Pervasive Computing 7(4): 12-18. doi: 10.1109/MPRV.2008.80

Haralick, M. and Shapiro, L.G. 1985. "Image segmentation techniques." *Computer Vision Graphics and Image Processing* 29: 100-132. doi: 10.1016/S0734-189X(85)90153-7

Harrie, L., H. Stigmar, T. Koivula, and L. Lehto. 2004. "An Algorithm for Icon Placement on a Real-Time Map." In *Developments in spatial data handling*, ed. P. Fisher. 493-507. Berlin: Springer.

Hirsch, S.A. 1982. "An Algorithm for Automatic Name Placement around Point Data." *American Cartographer* 9(1): 5-17. doi: 10.1559/152304082783948367

Hong, F., Z. Zuxun, and D. Daosheng. 2005. "Quality evaluation model for map labeling." *Geo-spatial Information Science* 8(1): 72-78. doi: 10.1007/BF02826996

Imhof, E. 1962. "Die Anordnung der Namen in der Karte." In *International Yearbook of Cartography*, 93-129, Bonn-Bad Godesberg: Kirschbaum.

Imhof, E. 1972. Thematische Kartographie. Berlin: Walter de Gruyter.

Imhof, E. 1975. "Positioning names on maps." *American Cartographer* 2(2): 128-144. doi: 10.1559/152304075784313304

Imhof, E. 1982. Cartographic Relief Presentation. Berlin: Walter de Gruyter.

Imhof, E. 2007. Cartographic Relief Presentation (reprint). Redmond: ESRI Press

Jenny, B., and L. Hurni. 2006. "Swiss-Style Colour Relief Shading Modulated by Elevation and by Exposure to Illumination." *Cartographic Journal* 43(3): 198-207. doi: 10.1179/000870406X158164

Johnson, D.S., and U. Basoglu. 1989. "The use of artificial intelligence in the automated placement of cartographic names." In *Proc. AutoCarto 9*: 225-230. Maryland.

Jones, C. B. 1989. "Cartographic Name Placement with Prolog." *IEEE Computer Graphics and Applications* 9(5): 36-47. doi: 10.1109/38.35536

Kakoulis, K.G. and I.G. Tollis., 2003. "A Unified Approach to Automatic Label Placement." *International Journal of Computational Geometry and Applications* 13(1): 23-59. doi: 10.1142/S0218195903001062

Kato, T., and H. Imai. 1988. "The NP-completeness of the Character Placement Problem of 2 or 3 Degrees of Freedom." Record of Joint Conference of Electrical and Electronic Engineers in Kyushu, 11-18.

Keates, J.S. 1973. Cartographic Design and Production. London: Longman.

Kern, J.R., and C.A. Brewer. 2008. "Automation and the Map Label Placement Problem: A Comparison of Two GIS Implementations of Label Placement." *Cartographic Perspectives* 60, 22-45.

Kirkpatrick, S., Jr.C.D. Gelatt, and M.P. Vecchi. 1983. "Optimization by Simulated Annealing." *Science* 220: 671-680. doi: 10.1126/science.220.4598.671

Klau, G.W., and P. Mutzel. 2003. "Optimal labeling of point features in rectangular labeling models." *Mathematical Programming* 94(2-3):435-458. doi: 10.1007/s10107-002-0327-9

Kovanen, J., and L.T. Sarjakoski. 2013. „Sequential displacement and grouping of point symbols in a mobile context." *Journal of Location Based Services* 7(2): 79-97. doi: 10.1080/17489725.2013.764024

Kraak, M.J. and A. Brown. 2001. Web Cartography – Developments and Prospects. London: Taylor & Francis.

Kraak, M.J. 2001. "Web resources and the map library." In *The Map Library in the New Millennium.* ed. R. B. Parry, and C.R. Perkins. 103-116. London : Library Association Publishing.

Laube, P., M. de Berg, and M. van Kreveld. 2008. “Spatial Support and Spatial Confidence for Spatial Association Rules.” In *Headway in Spatial Data Handling Lecture Notes in Geoinformation and Cartography.* ed. A. Ruas, and C. Gold. 575-593. Berlin Heidelberg: Springer. doi: 10.1007/978-3-540-68566-1_33

Lehmann, C., and J. Döllner. 2012. “Silhouette-Based Label Placement in Interactive 3D Maps.” *Cartography from Pole to Pole. Lecture Notes in Geoinformation and Cartography.* ed. M. Buchroithner, N. Prechtel, and D. Burghardt. 177-186. doi: 10.1007/978-3-642-32618-9_13

Li, Z., and P. Huang. 2002. “Quantitative measures for spatial information of maps.” *International Journal of Geographical Information Science* 16(7): 699-709. doi: 10.1080/13658810210149416

Lisle, R.J. 2003. Geological Structures and Maps: A Practical Guide. 3rd Edition. Oxford: Elsevier Butterworth-Heinemann.

Lloyd, R. 1997. “Visual Search Processes Used in Map Reading.” *Cartographica* 34(1): 11-32. doi: 10.3138/F342-1217-827R-0777

Lobben, A.K., and D.K. Patton. 2003. “Design guidelines for digital atlases.” Cartographic Perspectives 44: 51-62.

Loechel, A., and S. Schmid. 2013. “Comparison of Different Caching Techniques for High-Performance Web Map Services.” International Journal of Spatial Data Infrastructures Research 8: 43-73. doi: 10.2902/1725-0463.2013.08.art3

Lorena, L.A.N, and J.C. Furtado. 2001. “Constructive Genetic Algorithm for Clustering Problems.” *Evolutionary Computation* 9(3):309-327. doi: 10.1162/106365601750406019

Mackaness, W.A. “Automated Cartography in a Bush of Ghosts.” *Cartography and Geographic Information Science* 33(4): 245-256. doi: 10.1559/152304006779500678

Mairson, H.G., and J. Stolfi. “Reporting and Counting Intersections Between Two Sets of Line Segments.” *Theoretical Foundations of Computer Graphics and CAD*, Vol. 40, ed. R.A. Earnshaw. Springer Berlin Heidelberg. 307-325. doi: 10.1007/978-3-642-83539-1_11

MapSurfer.NET. 2014. A framework for map styling and rendering. Available at http://openmapsurfer.uni-hd.de/Framework/info.html

MapText. 2014. Label-EZ: Automated labeling software. Available at http://maptext.com/labelez.

Marks, J., and S. Shieber. 1991. “The computational complexity of cartographic label placement.” Technical Report TR-05-91, Harvard University, March.

Mooney, P., and P. Corcoran,. 2012. “Using OSM for LBS - An Analysis of Changes to Attributes of Spatial Objects.” In: *Advances in Location-Based Services.* 165-179. Berlin/Heidelberg, Germany. doi:10.1007/978-3-642-24198-7_11

Morrison, J.L. 1980. “Computer Technology and Cartographic Change.” In *The Computer in Contemporary Cartography*, ed. D.R.F. Taylor, 5-23. New York: Wiley.

Mote, K. 2007. “Fast Point-Feature Label Placement for Dynamic Visualizations.” *Information Visualization* 6(4): 249-260. doi: 10.1057/PALGRAVE.IVS.9500163

Neis, P., and A. Zipf. 2008. “OpenRouteService.org is Three Times “”Open”: Combining OpenSource, OpenLS and OpenStreetMaps.” In *Proc. of the GIS Research UK 16th Annual conference GISRUK 2008.* April 2-4 2008. Manchester.

New York Times. 2012. Facing Fees, Some Sites are Bypassing Google Maps. http://www.nytimes.com/2012/03/20/technology/many-sites-chart-a-new-course-as-google-expands-fees.html?pagewanted=all&_r=0. (accessed 25 March 2014).

Nöllenburg, M., and A. Wolff. 2011. “Drawing and Labeling High-Quality Metro Maps by Mixed-Integer Programming.” *Visualization and Computer Graphics, IEEE Transactions on* 17(5): 626-641. doi: 10.1109/TVCG.2010.81

Noyes, L. 1980. “The Positioning of Type on Maps: The Effect of Surrounding Material on Word Recognition Time.” *Human Factors: The Journal of the Human Factors and Ergonomics Society* 22(3): 353-360. doi: 10.1177/001872088002200309

Open Geospatial Consortium. 2006. Symbology Encoding Implementation Specification, Tech. Rep. 05-077r4. Ed. M. Müller. Available at http://portal.opengeospatial.org/files/?artifact_id=16700.

Open Geospatial Consortium. 2006. OpenGIS Web Map Server Implementation Specification. Tech. Rep. OGC 06-042, ed. J. de la Beaujardiere, Available at http://portal.opengeospatial.org/files/?artifact_id=14416.

Open Geospatial Consortium. 2007. Styled Layer Descriptor profile of the Web Map Service Implementations Specification. Tech. Rep. OGC 05-078r4, ed. M. Lupp, Available at http://portal.opengeospatial.org/files/?artifact_id=22364.

Open Geospatial Consortium. 2010. OpenGIS Web Map Tile Service Implementation Standard. Tech. Rep. OGC 07-057r7, ed. J. Masó, K. Pomakis, N. Julià. Available at http://portal.opengeospatial.org/files/?artifact_id=35326.

Open Geospatial Consortium. 2011. OpenGIS Implementation Standard for Geographic Information – Simple features access – Part 1: Common architecture, Tech. Rep. 06-

103r4. Ed. J.R. Herring. Available at http://portal.opengeospatial.org/files/?artifact_id=25355.

OpenMapSurfer. 2014. (a web map service provided by Heidelberg University; accessed August 10, 2014). Available at http://openmapsurfer.uni-hd.de.

PAL. 2014. A library for automated label placement (accessed on 20 August 2014). Available at http://pal.heig-vd.ch.

Pal, N.R., and S.K. Pal. 1993. "A review on image segmentation techniques." *Pattern Recognition* 29: 1277-1294. doi: 10.1016/0031-3203(93)90135-J

Peterson, M.P. 2003. "Maps and the Internet: Introduction." In *Maps and the Internet*, ed. M.P. Peterson. 1-16. Cambridge: Elsevier Press.

Peterson, M.P. 2008. "Trends in Internet and Ubiquitous Cartography". *Cartographic Perspectives* 61: 36-49.

Peterson, M. (ed.) 2012. Online Maps with APIs and Web Services. Berlin: Springer-Verlag.

Phillips, R.J., L. Noyes, and R.J. Audley. 1977. "The Legibility of Type on Maps." *Ergonomics* 20(6): 671-682. doi: 10.1080/00140137708931679

Phillips, R.J., L. Noyes. 1982. "An Investigation of Visual Clutter in the Topographic Base of a Geological Map." *Cartographic Journal* 19(2): 122-132. doi: 10.1179/caj.1982.19.2.122

Pinto, I. and H. Freeman. 1996. "The Feedback Approach to Cartographic Areal Text Placement." In *Advances in Structural and Syntactical Pattern Recognition, Lecture Notes in Computer Science* 1121: 341-350. New York: Springer. doi: 10.1007/3-540-61577-6_35

Plewe, B. 2007. "Web Cartography in the United Sates." Cartography and Geographic Information Science 34(2): 133-136. doi: 10.1559/152304007781002235

Rabello, R.L., G.R. Mauri, G.M. Ribeiro, and L.A.N. Lorena. 2014. "A Clustering Search Metaheuristic for the Point-Feature Cartographic Label Placement Problem." *European Journal of Operational Research* 234:802–808. doi: 10.1016/j.ejor.2013.10.021

Ramm, F., J. Topf, and S. Chilton. 2010. *OpenStreetMap: Using and Enhancing the Free Map of the World*. Cambridge: UIT Cambridge.

Raposo, P., C. Brewer, and L. Stanislawski. 2013. "Label and Attribute-Based Topographic Point Thinning." In *Proc. ICA Workshop on Generalisation and Multiple Representation*, August 25-30, Dresden, Germany.

Regnauld, N, S. Lessware, C. Wesson, and P. Martin. 2013. “Deriving Products from a Multiple Resolution Database using Automated Generalisation at Ordnance Survey.” In Proc. 26th International Cartographic Conference, August 25-30, Dresden, Germany.

Revell, P., N. Regnauld, and G. Bulbrooke. 2011. “OS Vectormap District: Automated Generalisation, Text Placement and Conflation in Support of Making Public Data Public.” In *Proc. 25th International Cartographic Conference*, July 3-8, Paris, France.

Ribeiro, G.M., G.R. Mauri, and L.A.N. Lorena. 2011. “A lagrangean decomposition for the maximum independent set problem applied to map labeling.” *Operational Researach* 11(3):229-243. doi: 10.1007/s12351-009-0075-1

Robinson, A.H., J.L. Morrison, P.C. Muehrcke, A.J. Kimerling, and S.C. Guptil 1995. *Elements of Cartography*, 6th ed. New York: Wiley.

Roth, R.E. 2013. “Interactive maps: What we know and what we need to know.” *Journal of Spatial Information Science* 6: 59-115. doi: 10.5311/JOSIS.2013.6.105

Rylov, M.A., and Reimer, A.W. 2014a. “A Comprehensive Multi-Criteria Model for High Cartographic Quality Point-Feature Label Placement.” *Cartographica* 49(1): 52-68. doi: 10.3138/carto.49.1.2137

Rylov, M.A., and Reimer, A.W. 2014b “A Practical Algorithm for the External Annotation of Area Features.” *Cartographic Journal*. doi: 10.1179/1743277414Y.0000000091

Rylov, M.A., and Reimer, A.W. 2014c “Improving Label Placement Quality by considering Basemap Detail with a Raster-Based Approach.” *GeoInformatica*. doi: 10.1007/s10707-014-0214-6

Rylov M.A., and A. Zipf. 2012. “Solutions for Limitations in Label Placement in OGC Symbology Encoding (SE) Specification.” In Proc. Geoinformatik 2012. Braunschweig, March 2012.

Schreyer, M., and G.R. Raidl. 2002. “Letting Ants Labeling Point Features.” In Proc. IEEE Congress on Evolutionary Computation (CEC '02) 2: 1564-1569.

Schrijver, S. 2003. Combinatorial Optimization: Polyhedra and Efficiency. Berlin: Springer.

Shamos, M., and D. Hoey. 1976. “Geometric Intersection Problems.” In *Proc. of the 17-th Ann. Conf. Foundations of Computer Science*: 208-215. doi: 10.1109/SFCS.1976.16

Shannon, C.E., and W. Weaver. 1964. “The Mathematical Theory of Communication.” Champaign: University of Illinois Press.

SRTM DGM: Free 90 meter CSI CIGAR version: http://*srtm*.csi.cgiar.org/

Stadler, G., T. Steiner, and J. Beiglbock. 2006. “A Practical Map Labeling Algorithm Utilizing Morphological Image Processing and Force-Directed Methods.” *Cartography and Geographic Information Science* 33(3): 207-215. doi: 10.1559/152304006779077327

Steiniger, S., and A.J.S Hunter. 2013. “The 2012 free and open source GIS software map – A guide to facilitate research, development and adoption.” *Computers, Environment and Urban Systems* 39: 136-150. doi: 10.1016/j.compenvurbsys.2012.10.003

Strijk, T.W., and M.J. van Kreveld. 2002. “Practical Extensions of Point Labeling in the Slider Model”, *GeoInformatica* 6(2): 181-197. doi: 10.1023/A:1015202410664

Sykora, P., O. Schnabel, I.I. Enescu, and L. Hurni. 2007. “Extended Cartographic Interfaces for Open Distributed Processing.” *Cartographica* 42(3): 209-218. doi: 10.3138/carto.42.3.209

Tobler, W.R. 1959. “Automation and Cartography.” *Geographical Review* 49(4): 526-534. doi: 10.2307/212211

Tsou, M.-H. 2011. “Revisiting Web Cartography in the United States: the Rise of User-Centered Design.” *Cartography and Geographic Information Science* 38(3): 250-257. doi: 10.1559/15230406382250

van Altena, V., R. Nijhuis, M. Post, B. Bruns, and J. Stoter. 2013. “Automated generalisation in production at Kadaster NL.” In *Proc. 26*th *International Cartographic Conference*, August 25-30, Dresden, Germany.

van Dijk, S. 2001. “Genetic Algorithms for Map Labeling.” PhD thesis, Department of Computer Science, Utrecht University, November.

van Dijk, S., M.J. van Kreveld, T.W. Strijk, and A. Wolff. 2002. “Towards an Evaluation of Quality for Names Placement Methods.” *International Journal of Geographical Information Science* 16(7): 641-661. doi: 10.1080/13658810210138742

van Dijk, S., D. Thierens, and M. de Berg. 2004. “On the Design and Analysis of Competent Selecto-Recombinative GAs.” *Evolutionary Computation* 12(2): 243-267. doi: 10.1162/106365604323142842

van Elzakker, C.P.J.M. 2000. “Use and Users of Maps on the Web.” *Cartographic Perspectives* 37: 34-50.

van Kreveld, M.J., T.W. Strijk, and A. Wolff. 1999. “Point Labeling with Sliding Labels.” *Computational Geometry: Theory and Applications* 13: 21-47. doi: 10.1016/S0925-7721(99)00005-X van Roessel, W. 1989. “An Algorithm for Locating Candidate Labeling Boxes Within a Polygon.” *The American Cartographer* 16(3): 201-209. doi: 10.1559/152304089783814034

Verner, O.V., R.L. Wainwright, and D. Schoenefeld. 1997. “Placing Text Labels on Maps and Diagrams using Genetic Algorithms with Masking.” *INFORMS Journal on Computing* 9(3): 266-275. doi: 10.1287/ijoc.9.3.266

Wagner, F., A. Wolff, V. Kapoor, and T. Strijk. 2001. “Three Rules Suffice for Good Label Placement.” *Algorithmica* 30(2): 334-349. doi:10.1007/s00453-001-0009-7

Wardlaw, J. 2010. “Principles of interaction.” In *Interacting with geospatial technologies.* ed. M. Hacklay. 179-198, Wiley-Blackwell, West Sussexx, UK. doi: 10.1002/9780470689813.ch9

Wein, R. 2007. “Exact and approximate construction of offset polygons.” *Computer-Aided Design* 39(6): 518-527. doi: 10.1016/j.cad.2007.01.010

Wolff, A., L. Knipping, M.J. van Kreveld, T.W. Strijk, and P.K. Agarwal. 2001. “A Simple and Efficient Algorithm for High-Quality Line Labeling.” *Innovations in GIS VII: GeoComputation* 11: 147-159.

Wolff, A., and T. Strijk, 2009. “A map labeling bibliography.” Available at http://i11www.iti.uni-karlsruhe.de/~awolff/map-labeling/bibliography/.

Wood, C.H. 1994. “Effects of Brightness Difference on Specific Map-Analysis Tasks: An Empirical Analysis.” *Cartography and Geographic Information Science* 21(1): 15-30. doi: 10.1559/152304094782564019

Wood, C.H. 2000. “Descriptive and Illustrated Guide for Type Placement on Small Scale Maps.” *Cartographic Journal* 37(1): 5-18. doi: 10.1179/000870400787320851

Yamamoto, M., G. Camara, and L.A.N. Lorena. 2002. “Tabu Search Heuristic for Point-Feature Cartographic Label Placement.” *Geoinformatica* 6(1): 77-90. doi: 10.1023/A:1013720231747

Yang, C., D.W., Wong, R. Yang, M. Kafatos, and Q. Li. 2005. “Performance-improving techniques in web-based GIS.” *International Journal of Geographical Information Science* 19(3): 319-342. doi: 10.1080/13658810412331280202

Yoeli, P. 1972. “The Logic of Automated Map Lettering.” *Cartographic Journal* 9(2): 99-108. doi: 10.1179/000870472787352505

Zhang, Q. and L. Harrie. 2006. “Placing Text and Icon Labels Simultaneously: A Real-Time Method.” *Cartography and Geographic Information Science* 33(1): 53-64. doi: 10.1559/152304006777323127

Zighed, J.A., R. Abdesselam, and A. Hadgu. 2012. "Topological Comparisons of Proximity Measures", In *Advances in Knowledge Discovery and Data Mining Lecture Notes in Computer Science* 7301: 379-391. doi: 10.1007/978-3-642-30217-6_32

Zoraster, S. 1986. "Integer Programming Applied to the Map Label Placement Problem." *Cartographica* 23(3): 16-27. doi: 10.3138/9258-63QL-3988-110H

Zoraster, S. 1990. "The Solution of Large 0-1 Integer Programming Problems Encountered in Automated Cartography." *Operations Research* 38(5): 752-759. doi: 10.1287/opre.38.5.752

Zoraster, S. 1997. "Practical Results using Simulated Annealing for Point Feature Label Placement." *Cartography and Geographic Information Science* 24(4): 228-238. doi: 10.1559/152304097782439259

Part II.

Publications

5 A Comprehensive Multi-Criteria Model for High Cartographic Quality Point-Feature Label Placement

Authors	Maxim A. Rylov and Andreas W. Reimer
Journal	*Cartographica: The International Journal for Geographic Information and Geovisualization*
History	Published: 26 March 2014 Accepted: 11 November 2013 Revised: 25 November 2013 Submitted: 05 August 2013
Reference	Rylov, M.A., and A.W. Reimer. 2014. “A Comprehensive Multi-Criteria Model for High Cartographic Quality Point-Feature Label Placement.” *Cartographica* 49(1): 52-68. doi: 10.3138/carto.49.1.2137

5 A Comprehensive Multi-Criteria Model for High Cartographic Quality Point-Feature Label Placement

Abstract

The cartographic lettering process is an essential part of geographic map production. Assigning names to point-features is one of the map lettering tasks. There have been numerous and varied research efforts to automate point-feature label placement (PFLP). It seems that none of them take into account the many well-established cartographic precepts for point-feature annotation used by human cartographers. As a consequence, the currently implemented, fully automated solutions are limited in their expressive power. Hence, the PFLP problem is still vital, and its solution is a compelling challenge. In this paper we present a comprehensive multi-criteria model that complies with almost all well-defined cartographic placement principles and requirements for PFLP. This allows for a significant increase in toponym density without effecting legibility. The proposed model expressed as a quality evaluation function can be employed by any mathematical optimization algorithm for solving the automated label placement problem. An application of the proposed model was tested on Volunteered Geographic Information data and sample parameter settings were devised. The results illustrate that a high level of cartographic quality for PFLP can be achieved through the integrated approach. The resulting quality is comparable to the lettering produced by an expert cartographer.

Keywords: automated cartography, automated label placement, combinatorial optimization, GIS mapping

5.1 Introduction

Maps convey spatially distributed geographic information to a reader. As a medium of communication maps should be clear and legible. Good name positioning is a key factor of cartographic representation (Imhof, 1972; Bertin, 1983; Robinson et al., 1995) and communication. Manual map labelling is a tedious and time-consuming task. Morrison (1980) estimated that manual lettering can take up to 50% of total map production time. Over the last four decades many attempts have been made to automate the process of map labelling; see bibliography of papers on this topic maintained by Wolff and Strijk (2009). The main reason is to reduce the cost of manual label placement on the map.

The cartographic lettering problem comprises (e.g., Imhof, 1972; Keates, 1973; Robinson et al., 1995):

- questions regarding toponyms (e.g., exonyms vs. endonyms, colloquial place names, multilingual labelling, and gazetteers or place-name databases);
- toponym selection (quantity and type) in respect to intended map-use task;
- questions regarding typeface (e.g., font choice, font form (spacing, colouring, italics etc.), font size, semantic systematization of typeface related choices);
- geometric placement of labels.

Automated label placement research has concentrated on the geometric placement and assumes the preceding problems have been solved, i.e. are input parameters. Three classes of label placement tasks are commonly identified both in automated and manual lettering:

- labelling of point like objects (e.g., settlements, mountain peaks);
- linear features (e.g., roads, boundaries, rivers);
- areal features (e.g., countries, islands, lakes, mountain ranges).

Cartographers (Imhof, 1962, 1975; Wood, 2000; Brewer, 2005) summarized and formulated a series of precepts about effectively placing text on maps for each task. The main difficulty of automating these cartographic rules is to quantify them through a unique measure. However, a group of researchers (van Dijk et al., 2002) managed to formalize and classify most requirements of good label placement and developed a general function that numerically measures the quality of label placement. van Dijk et al. (2002) stated that their function is a first step towards such a formalization in which the implementation of the partial functions is still required.

Most previous research efforts in the field of automated label placement focused on point-feature labelling as a geometric problem. As their main goal was to place the maximum number of labels on the map without any overlap, they omitted other rather important aspects such as aesthetics, label-feature association or feature visibility (Figure 5.1). Only a few methods suggested in the literature consider the formalized cartographic guidelines for these essential properties. The main strategy that is employed to circumvent these deficiencies in web-mapping and automated map production for national napping agencies (NMAs) is thinning out map content (Figure 5.1a). Such sparse lettering is employed by Google for Google Maps (Figure 5.1a), Mapnik for *OpenStreetMap* (OSM) (Figure 5.10) (Haklay and Weber, 2008; Ramm et al., 2010), the USGS (Raposo et al., 2013), Dutch Kadaster (van Altena et al., 2013), and the Ordnance Survey (Revell et al., 2011; Regnauld et al., 2013) for automated static map production. This limits the use of automation to contextual/background maps in contrast to a full inventory for a given scale. Furthermore, the thinning techniques need to be developed for each agency and dataset separately as the USGS and Ordnance Survey examples show. A more powerful labelling model, as presented in this paper, can potentially aid map procedures in overcoming limitations of the related work and commercial packages.

(a)

(b)

Figure 5.1. Map of Sicily. (a) Automated label placement. Source: Map data ©2013 Google. (b) Manual lettering. Source: Encyclopedia Britannica World Atlas (1963), map © RM Acquisition, LLC d/b/a Rand McNally. Reproduced with permission, License No. R.L. 14-S-002. All rights reserved.

In this paper we summarize the most relevant requirements for point-feature label placement (PFLP) and construct a comprehensive model that encapsulates most cartographic requirements in one single formula. The requirements that remain unaddressed are those concerning the semantic and topological relationship of a feature with its surrounding. For example, for a settlement to the left of a river, place its name also entirely to the left of that line. The suggested approach does take into account all purely visual relationships between features and their surrounding/background. Another restriction for the presented approach is that we only concern ourselves with axis-aligned labelling. Our model is expressed as a quality function computed as the weighted sum of simple metrics that correspond to the set of established cartographic rules for PFLP. Some of these metrics were formalized and classified, but not yet expressed through analytic formulas or were only described sparsely in the literature (see Section 5.3.2, Table 5.1). They include a metric for coastal places and a measure of label cluttering or measure for the degree of ambiguous association between adjacent labels. It is notable that the proposed quality function can be employed by any mathematical optimization algorithm for solving the PFLP problem.

We start by giving a short review of the related work in the field of automated label placement (Section 5.2). Then we define the PFLP problem, write out cartographic requirements for it, and formally define our model (Section 5.3). In Section 5.4 we present some results of our experiments that were performed on various PFLP algorithms (Christensen et al., 1995) and with different input parameters of the model. The experiments were carried out on the dataset based on *Volunteered Geographic Information* (VGI; Goodchild, 2007), namely on OSM geodata. The resulting maps illustrate the advantages of the proposed multi-criteria model. Finally, we conclude with an analysis of the model, discuss open questions, and directions for future work (Section 5.5).

5.2 Related Work

In this section, we review the most significant techniques, models and optimization strategies that were invented to automate the process of label placement in the field of cartography. The short review summarizes previous research that serves as a basis for our model and helped us develop and test our model.

The history of automated label placement started from the work of Imhof (1962) and Imhof (1975). Imhof's broad guidelines for positioning names on maps stimulated the development of various algorithms and rule-based "expert" systems (Ahn and Freeman, 1984; Hirsch, 1982; Yoeli, 1972) that automatically place labels on maps, graphs or diagrams. In his work, Imhof provided general principles and requirements that guide name placement for three types of designations. These categories are point, linear and areal designations. The task of lettering a certain type of designation has its own requirements and involves its own challenges. In automated label placement the partition of the tasks is preserved.

The most attention and comprehensive study in previous work focused on the problem of labelling point features. The PFLP problem requires placement of labels adjacent to point features in such a way that overlap of labels is minimized or equals zero. The complexity analysis has shown that basic PFLP is an *NP-hard* problem. This fact has been proved by different research groups independently, first by Kato and Imai (1988) and later by Marks and Shieber (1991) and Formann and Wagner (1991).

Many attempts and compelling techniques were suggested over the last decades to solve the PFLP problem. The first algorithm, presented in the early 1970s by Yoeli (1972), used a *depth-first search* approach. Then Hirsch (1982) applied a *discrete gradient descent* method. Those inferior techniques were just a first step in developing more mature and sophisticated labelling algorithms. In 1986 and 1990 Zoraster used a variant of *0-1 integer programming* to reduce the PFLP problem. At the same time other researchers were using exhaustive search algorithms (Ahn and Freeman, 1984; Freeman and Ahn, 1987; Jones, 1989; Cook and Jones, 1990; Doerschler and Freeman, 1992). The idea of using the *simulated annealing* algorithm (Kirkpatrick et al., 1983) for solving the PFLP problem was pursued by Christensen et al. (1995) and Zoraster (1997). Further, a *genetic* algorithm for the PFLP problem was proposed by Verner et al. (1997). Yamamoto et al. (2002) applied a *tabu search* heuristic to cartographic label placement. Schreyer and Raidl (2002) presented a new effective approach that is based on the concept of *ant colony* optimization. After that period there were a couple of attempts to improve the performance and quality of existing algorithms. For instance, Ebner et al. (2003) developed a *force-based simulated annealing* algorithm that uses repulsive forces between labels. van Dijk et al. (2004) designed and analyzed the competent *selecto-recombinative* approach for a genetic algorithm. Stadler et al. (2006) proposed a novel approach for automated map labelling that combines a discrete method based on image processing ideas and a continuous *force-directed* method. Inspired by the requirements of dynamic and interactive maps, some researchers presented their refined techniques to solve the PFLP problem in real-time. Mote (2007) proposed a novel geometric *"de-confliction"* approach that is distinguished from competitors by speed and scalability. Later, Luboschik et al. (2008) presented a *particle-based* approach that performs very fast non-overlapping PFLP and respects other visual elements of a map. The most recent paper, by Gomes et al. (2013), addresses the discrete dispersion concept, which has an objective to maximize the minimum distance between labels.

Two different labelling models emerged; the *fixed position* and *sliding* label models. It was demonstrated that a simple implementation of the sliding model (van Kreveld et al., 1999; Strijk and van Kreveld, 2002) was able to outperform simulated annealing (Christensen et al., 1995) by up to 10% in the number of labels placed. The explanation is very simple. The sliding model allows labels to be placed anywhere around the point, thus granting additional freedom in placement by increasing the search space.

Most algorithms for PFLP use the simplest guidelines from a rich set of guidelines provided by Imhof (1962/1975). More exactly, the constraint is that names should not overlap each other. In other words, the objective was to maximize the number of labels without overlap (Klau and Mutzel, 2003) or minimize the area of overlap between labels. However, at some point, the task of evaluating label positions, according to the cartographic guidelines, became more important. Edmondson et al. (1996) proposed to compute a single numeric score by using a function as a weighted sum of simple metrics. In their work Edmondson et al. (1996) described two metrics: (1) spatial crowding and overlap and (2) positioning. Later van Dijk et al. (2002) formalized most label placement rules into a quality evaluation function. Their function covers four aspects of label placement quality. These are aesthetics, label visibility, feature visibility, and label-feature association. Their model achieved two main goals: to provide a numeric evaluation of the cartographic rules, and to compare how well different labelling algorithms perform their tasks.

Research in the field of automated label placement over the past 40 years has produced significant methodology for furthering the theoretic development of PFLP. Using this with a comprehensive model for cartographic principles relevant to map production, should allow automated labelling to approach the quality of manual labelling done by skilled cartographers.

5.3 Multi-Criteria Model

In this section we give a detailed description of our methodology. First we define the cartographic PFLP. Then we summarize a list of exacting cartographic rules for PFLP which we use in constructing a model to evaluate the quality of labelling. For each cartographic rule we define a quality metric and describe it in detail.

5.3.1 Definition of the PFLP

The PFLP task assigns labels to a set of point features. Among different techniques used for solving the PFLP problem, we have chosen an approach that was first suggested by Cromley (1985) and Zoraster (1986). These authors considered PFLP as a problem of *combinatorial optimization*. Thus, a mathematical programming technique can be applied to solve the PFLP problem (Christensen et al. 1995; Huffman and Cromley, 2002). The main advantage of mathematical optimization algorithms is that they are easily implemented in programming languages. Each combinatorial optimization problem (COP) consists of two components: *a search space* and *an objective function*. In the case of PFLP search space can be defined as follows:

Suppose we are given a set of N points in the Euclidean plane. Each point describes a point feature that needs to be labelled. The label's position is chosen from a set of potential label positions P_n, also known as candidate positions (Figure 5.2). Since, we can define potential label positions for all point features as:

$$x_{i,j}, \forall\ i = 1, \dots, P_n, \forall\ j = 1, \dots, N \tag{5.2}$$

where $x_{i,j} \in \{0,1\}$ is the decision variable that defines whether the jth point feature is labelled in the ith position. The requirement that a point feature can be labelled only once or even can be left unlabelled can be written as:

$$\sum_{i=1}^{P_n} x_{i,j} \leq 1, \forall\ j = 1, \dots, N. \tag{5.3}$$

An important requirement, or even a constraint, of the map lettering is that no two labels may partly or completely overlap each other or the point features. This requirement can be formulated in the following inequality:

$$x_{i,j} + x_{k,m} \leq 1 \tag{5.4}$$

which is valid for all intersections between the ith position of the jth point feature and the kth position of the mth point feature.

Now let us consider the second aspect of the COP. The objective function measures the quality of a label placement. The general form of the objective (quality) function is written as:

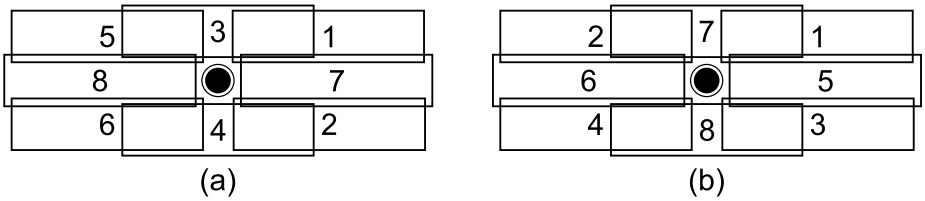


Figure 5.2. A model for positional prioritization of point-feature labelling: (a) Imhof's variant (Imhof, 1962, 1975); (b) Yoeli's version (Yoeli, 1972).

$$\text{Maximize } Q(x) = \sum_{i=1}^{P_n} \sum_{j=1}^{N} w_{i,j} x_{i,j} \tag{5.5}$$

where $x = (x_{1,1}, \dots, x_{P_n,N})$ is a vector that represents a complete set of decision variables for the set of potential labels $L = \{l_{1,1}, \dots, l_{P_n,N}\}$ for N point features, and $w_{i,j}$ are weighting parameters (metrics) that represent cartographic preferences of label placement (Zoraster, 1986; Christensen et al., 1995). The goal of the COP is to find an optimal feasible solution within a search space (5.2) that maximizes or minimizes (the choice is arbitrary) an objective function (5.5) by taking into account constraints (5.3) and (5.4).

We note that in our implementation we use the eight-position model for which $P_n = 8$. The *fixed position* model of eight standard positions is typically used in cartography. However, the ordering of preferable positions is rather diverse in the literature. Wu and Buttenfield (1991) performed a detailed study addressing the validity of a certain prioritization model used by different publishing houses for labelling of road maps.

5.3.2 PFLP Rules and Constraints

Cartographers refined rules for good label placement over many years (Imhof, 1962, 1975; Wood, 2000; Brewer, 2005). We studied certain broad cartographic guidelines in the literature that refer to labelling of point features. Those guidelines can be divided into two categories: rules and constraints. The list of rules adapted to the requirements of our purposes in the PFLP problem is as follows:

R1. Type arrangement should reflect the classification, importance and hierarchy of objects.
R2. Labels should be placed horizontally.
R3. Placement of the lettering to the right and slightly above the symbol is prioritized.
R4. Names of coastal settlements should be written on the water.
R5. Labels should be placed completely on the land or completely on the water surface.

R6. Names should not be too close to each other.

R7. Labels should not be excessively clustered nor evenly spread out.

R8. Each label should be easily identified with its point feature. Ambiguous relationships between symbols and their names must be avoided.

R9. Labels should not overlap other significant features of the map background or do this as little as possible.

The list of constraints is:

C1. Names must not overlap point feature symbols.

C2. Two names must not overlap each other.

C3. Two point features must not have any overlap.

The constraints C1–C3 are used in the generation of a search space. The way candidate positions are generated (Figure 5.2) satisfies the constraint C1. C2 and C3 are fulfilled automatically due to inequality (5.4). The list of requirements R1–R9 is used as the criterion for constructing the metrics in the objective function (5.5).

Before moving forward we want to understand how well the given rules were carried out in the previous works. A more detailed and comprehensive guide of cartographic rules for all label types is provided in the work by van Dijk et al. (2002). It consists of approximately sixty criteria relevant to good label placement which are classified into four groups. They are aesthetic quality, label visibility, feature visibility, and label-feature visibility. This classification was used in a review paper by Kern and Brewer (2008), who created a table which contains van Dijk's groups of criteria (van Dijk et al., 2002) to make a particular comparison of various labelling techniques and algorithms presented in the literature. Among the ~60 criteria collected by van Dijk et al. (2002), only 13 are concerned with point-feature labelling. Our rules and constraints cover nine of these. Of the remaining four, the first (i.e. placement at the space between parts of compound words) is addressed via our implementation, the second is concerned with curved labels (Section 5.1), and the third is concerned with semantic constraints (Section 1). The fourth missing criterion, "Do not intersect point labels with a grid line", we deem inappropriate for cartography.

To show the contribution of our approach we tabled the most significant PFLP models and algorithms from the literature according to our lists of rules and constraints (Table 5.1). It can be seen from Table 5.1 that none of the other preceding techniques covers all cartographic requirements and constraints for the PFLP problem. Table 5.1 indicates that the most evaluated requirements are R1, R2, and R3. Furthermore, almost all algorithms comply with the constraints C1–C3 with three exceptions: the methods by Ebner et al. (2003), Bae et al. (2011), and Gomes et al. (2013) which address the issue of minimizing label overlap. At the same time only half of the authors pursue the objective of R9. Besides, we can see that R6 and R7 are rather poorly taken into account. The rest of the rules, R4, R5, and R8, have not been studied yet in the literature. The lack of coverage for R4–R8 is not interpreted as an absolute shortcoming of the related work, but as a relative shortcoming from the perspective of map production. The related work generally approached the PFLP as a problem within the realm of computational geometry (CG) and succinctly defined it in that spirit. The CG-PFLP is defined by placing as many labels as possible without overlaps and with some rules to constrain the search space. From a cartographic perspective, the PFLP has a different emphasis: to unambiguously place as few labels as needed for the map product.

Placement rules from the cartographic literature are not mainly about aesthetics in the sense of art, beauty and taste as implied by Formann and Wagner (1991). Rather, they are focused on disambiguation to support the topographic and semantic content of the map (Imhof, 1962/1972/1975; Anson, 1988; Robinson et al., 1995, etc.). Our multi-criteria model aims to fill this gap between sound computational theory and cartographic

Table 5.1. The PFLP requirements and constraints in previous research works.

Research article	PFLP requirements and constraints applied											
	R1	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**C1**	**C2**	**C3**
Yoeli (1972)	X	X	X							X	X	X
Hirsch (1982)	X	X	X						X	X	X	X
Zoraster (1986)		X	X							X	X	X
Doerschler&Freeman (1992)	X	X	X						X	X	X	X
Edmondson et al. (1996)	X	X	X						X	X	X	X
Strijk & van Kreveld (2002)	X	X							X	X	X	X
Huffman & Cromley (2002)	X	X	X							X	X	X
Ebner et al. (2003)		X	X			X	X					
van Dijk et al. (2004)		X								X	X	X
Stadler et al. (2006)	X	X				X			X	X	X	X
Mote (2007)	X	X	X							X	X	X
Luboschik et al. (2008)	X	X	X						X	X	X	X
Bae et al. (2011)		X	X									
Gomes et al. (2013)		X	X			X	X					

Table 5.2. The PFLP requirements and constraints in the presented model.

	R1	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**C1**	**C2**	**C3**
	X	X	X	X	X	X	X	X	X	X	X	X

requirements for disambiguation and topographic structure. In our approach we address all rules (R1–R9) and constraints (C1–C3) simultaneously (Table 5.2). Table 5.1 confirms that this task was not achieved before.

5.3.3 Quality Evaluation Function

In Section 5.3.1 we have defined a general form of the objective function (5.5). In the previous section we have also written out the requirements for good label placement. These requirements are also metrics for the quality evaluation function. First two basic factors that affect the quality of a name positioning must be defined. These factors, which represent cartographer's goals, are:

- Number of point features that are labelled (Huffman and Cromley, 2002).
- Cartographic preferences for a certain candidate position.

Let us combine these two factors into a single formula:

$$Q(x) = \alpha_1 Q_{\mathrm{ln}}(x) + \alpha_2 Q_{\mathrm{cp}}(x) \tag{5.6}$$

where x is defined in (5.2) and (5.5) and α_1, α_2 are weighing parameters. The function $Q_{\mathrm{ln}}(x)$ evaluates the percentage of point features that are labelled:

$$Q_{\mathrm{ln}}(x) = N_{\mathrm{l}}(x)/N \tag{5.7}$$

where $N_{\mathrm{l}}(x)$ is a function which returns the number of point features that were labelled. This function has the form:

$$N_{\mathrm{l}}(x) = \sum_{i=1}^{P_n} \sum_{j=1}^{N} x_{i,j} \tag{5.8}$$

The function $Q_{\mathrm{cp}}(x)$ in (5.6) is a quality function of cartographic preferences. It has the full form:

$$\begin{aligned} Q_{\mathrm{cp}}(x) = \Bigg[\sum_{i=1}^{P_n} \sum_{j=1}^{N} \big(& \beta_1 F_{i,j}^{\mathrm{prior}} + \beta_2 F_{i,j}^{\mathrm{pos}} + \beta_3 F_{i,j}^{\mathrm{over}} \\ & + \beta_4 F_{i,j}^{\mathrm{disamb}} + \beta_5 F_{i,j}^{\mathrm{clut}} + \beta_6 F_{i,j}^{\mathrm{coast}} \big) x_{i,j} \Bigg] / N_{\mathrm{l}}(x) \ . \end{aligned} \tag{5.9}$$

In equation (5.9), $F_{i,j}^{*}$ are the quality metrics:

$F_{i,j}^{\mathrm{prior}}$ – priority of the point feature;

$F_{i,j}^{\mathrm{pos}}$ – position of the name around its point feature in terms of cartographic desirability;

$F_{i,j}^{\mathrm{over}}$ – overlapping of symbol and its label with other significant map features;

$F_{i,j}^{\mathrm{disamb}}$ – magnitude of ambiguity between neighbouring point features and their names;

$F_{i,j}^{\mathrm{clut}}$ – reflects proximity of placed labels to each other;

$F_{i,j}^{\mathrm{coast}}$ – percentage of water under the label for the point features that describe coastal places;

where $\beta_1, \ldots, \beta_6$ are weights that define the contribution of each metric $F_{i,j}^{*}$ in terms of the total quality value. Each metric $F_{i,j}^{*}$ corresponds to one of the cartographic rules and quantifies the quality of a label for the jth point feature in the ith position. A detailed description of the metrics $F_{i,j}^{*}$ is provided in the following sub-sections. Note that parameters $\alpha_k, \beta_m, k = 1,2, m = 1, \ldots, 6$ should sum up to 1 by k and m respectively. Moreover, in order to get a normalized total quality value that belongs to the interval [0,1] we require that two functions Q_{ln}, Q_{cp} and the metrics $F_{i,j}^{*}$ also return values in the range [0,1]. The approach of adjustable weights affords the opportunity to prefer one cartographic rule over another, or even considers (5.6) as a pure *label number maximization problem* (Klau and Mutzel, 2003) by setting $\alpha_2 = 0$. Generally speaking, the adjusting of the weights should be done by the user.

5.3.4 Quality Metrics

5.3.4.1 Priority

In respect to the semantic communication goals of a map, place names should be printed in different sizes, as well as corresponding symbols. For example, the font type and size of the label for Frankfurt am Main (pop. ~690,000) should be different from the ones that are used to assign a label to Heidelberg (pop. ~150,000). The difference in presentation of two cities helps a reader to see the difference in population, importance, or administrative status of a place (Butzler et al., 2011). Such differentiation can be done by assigning a priority to a place. Let us define a metric as:

$$F_{i,j}^{\mathrm{prior}} = \frac{p_j - p_{\min}}{p_{\max} - p_{\min}} \tag{5.10}$$

where $p_j, j = 1, \ldots, N$ is a value of the priority of the jth point feature, $p_{\min}$ and $p_{\max}$ are minimum and maximum priority values of all point features on a map respectively, namely:

$$\begin{aligned} p_{\min} &= \min_{j \in N} p_j \\ p_{\max} &= \max_{j \in N} p_j. \end{aligned} \tag{5.11}$$

Equation (5.10) returns normalized values in the range [0,1]. The metric $F_{i,j}^{\mathrm{prior}}$ fulfills the requirement of R1. The simple linear model of the score function that we devised can also be replaced by any other; such as a quadratic model or fading function.

5.3.4.2. Positioning

The following metric concerns the next cartographic rule, R3. Following one of the guidelines of Imhof (1962, 1975) we can number positions around a point feature according to their desirability (Figure 5.2). Therefore, we determine the quality metric as:

$$F_{i,j}^{\mathrm{pos}} = 0.5 + \frac{0.5(P_n - i)}{P_n - 1} \tag{5.12}$$

where P_n is the total number of candidate positions around a point feature j, i is the sequence number of a candidate position (Figure 5.2). The metric has the maximum value when the label position is somewhat above and to the right of its symbol. We also should note that the way of generation of potential label positions meets the requirement of R2.

5.3.4.3 Feature Overlap

Having one name overlap another is not permissible in cartography (C1–C3), but by minimizing the degree of cartographic disturbance (R9), labels are allowed to overprint, cover, or even completely conceal other surrounding important geographic features (e.g., roads, rivers).

In early research, two approaches were suggested and used to measure the influence of labels on the features visibility: *vector-based* (Freeman and Ahn, 1984; Edmondson et al., 1996; Strijk and van Kreveld, 2002; Luboschik et al., 2008) and *raster-based* methods (Doerschler and Freeman, 1992; Harrie et al., 2004; Stadler et al., 2006; Luboschik et al., 2008). Both of them have advantages and disadvantages. The vector-based approach is fast, robust and efficient for maps with moderate density of features (large scales). On the other hand, it meets insurmountable difficulties when map density becomes very high, which discouragingly decreases the performance of any labelling algorithm. This approach does not take into account the cartographic appearance of lines (width, stroke, cap, etc.) or polygons (hatching) directly without extra computations. Instead, feature density has no influence on the raster-based method and considers the appearance of cartographic features, but it also has a pitfall. The problem is in how to measure the direction in which a label and a line feature overlap each other, for example, a line running the length or the width of the name. Another problematic aspect for the raster-based method arises due to the absence of the actual geometry of the mapped features: when defining whether or not the point feature and its name lie on the same side of the line feature next to the label.

Nevertheless, based upon characteristics of each approach, we have chosen the raster-based method. We prefer this method as it can be used on both small and large scales. Therefore, we define a metric that can measure *homogeneity of the map background* under a label. As an input for this metric we require a raster image I in which the non-textual objects are already rendered. Each element $p \in I$ is a pixel. Assume that we applied some image segmentation algorithm φ (Haralick and Shapiro, 1985) which transforms pixels into clusters $\varphi(I) = \{c^1, c^2, \dots, c^M\}$, where $c^m, m = 1, \dots, M$ are the clusters. Hence, each pixel $p \in I$ has the associated cluster index c_p. Next, we define the set of axis-aligned rectangles that bound the characters of the name of the jth point feature on the ith position as $R_{i,j} = \left\{r_{i,j,1}, \dots, r_{i,j,K_j}\right\}$, where K_j is the number of characters in the name of the jth point feature. We also demand that the coordinates and size of the rectangles are rounded to the pixel coordinates. In order to shorten further mathematical manipulations we denote all pixels that lie within a certain rectangle $r_{i,j,k}$ as $D_{r_{i,j,k}}$, $k = 1, \dots, K_j$. Next, let us define a metric that calculates the number of elements of the cluster with index c_m within $R_{i,j}$ as follows:

$$H_{i,j}(c_m) = \sum_{k=1}^{K_j} \sum_{q \in D_{r_{i,j,k}}} B(c_q, c_m) \tag{5.13}$$

where function B is defined as:

$$B(d, e) = \begin{cases} 1, & d = e \\ 0, & d \neq e \end{cases} \tag{5.14}$$

and d, e are cluster indices.

Hence, using (5.13) the background homogeneity metric can be written as:

$$F_{i,j}^{\text{over}} = \frac{\max_{m \in M} H_{i,j}(c_m)}{A(R_{i,j})} \tag{5.15}$$

where $A(R_{i,j})$ is the area of all rectangles in $R_{i,j}$. The metric $F_{i,j}^{\text{over}}$ returns a value in the interval [0,1]. This metric is designed to yield a value of 1.0 for the case when all elements within $R_{i,j}$ belong to one cluster, i.e. the region of the map background under a label is homogeneous.

5.3.4.4 Disambiguation

R8 states that a cartographer must strive to avoid any ambiguous relationships of symbols to names in the process of map lettering. This means that the ambiguity between two adjacent labels exists if they are too close to each other (R6). We consider two aspects that have an influence on ambiguity. They are:

- Two different labels should not be close to each other.
- Labels of different features that are close to each other should not be vertically or horizontally aligned.

The degree of proximity can be expressed as the geometric distance between a set of points that describe the geometry of a certain symbol and its name. We define the metric of *disambiguation* as a function of the distance that consists of *two parts*, see (5.19) and (5.20). Each part reflects an aspect that we defined above. At first we make some definitions.

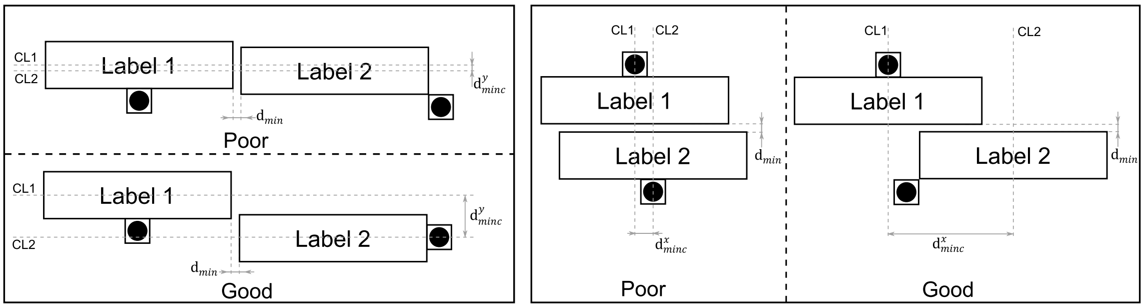
For every label $\eta \in L$ we define its bounds as $r_\eta = r_\eta^s \cup r_\eta^n$, where r_η^s and r_η^n are two rectangles on the Euclidean plane $\mathbb{R}^2$, which are minimum bounding rectangles of the symbol and its name respectively. Further, for every two labels $\eta, \mu \in L$ we define a function that returns the Euclidean distance between them:

$$d_{\min}(\eta,\mu) = \begin{cases} 0, & \eta \text{ and } \mu \text{ overlap} \\ \min\{d_{\text{minr}}(r_\eta^s, r_\mu^s), d_{\text{minr}}(r_\eta^s, r_\mu^n), & \text{otherwise} \\ \quad d_{\text{minr}}(r_\eta^n, r_\mu^s), d_{\text{minr}}(r_\eta^n, r_\mu^n)\} & \end{cases} \tag{5.16}$$

where $d_{\text{minr}}(r_1, r_2) = \min\{\|p,q\| \mid p \in r_1, q \in r_2\}$ is a minimum distance between two rectangles. The norm $\|p,q\|: \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ denotes the Euclidean distance between two points $p, q \in \mathbb{R}^2$. We also define $d_{\text{minc}}^x(r_1, r_2)$ and $d_{\text{minc}}^y(r_1, r_2)$ as functions that return the distance between x- or y-components of the centres of two rectangles r_1 and r_2 (Figure 5.3). We define the first one as:

$$d_{\text{minc}}^x(r_1, r_2) = \begin{cases} T_{\text{dc}}, & d_{\text{minr}}(r_1, r_2) \ge T_{\text{d}} \\ |x_{r_1} - x_{r_2}|, & d_{\text{minr}}(r_1, r_2) < T_{\text{d}} \end{cases} \tag{5.17}$$

where x_{r_1}, x_{r_2} are the x- components of the centre points of r_1 and r_2 respectively. Note, that function $d_{\text{minc}}^x(r_1, r_2)$ returns a threshold value T_{dc} in the case when two rectangles are far enough apart to create any ambiguities between two labels. The parameter T_{dc} is the threshold distance between the horizontal or vertical centrelines of two rectangles



(a) (b)

Figure 5.3. Presentation of ambiguity between two neighbouring labels. Note that the cases which qualified as poor are acceptable in final labelling if there is no other space for better label placement (i.e. it is a soft constraint).

that forms the bounds of the labels (see *CL1* and *CL2* in Figure 5.3). The function $d^y_{\mathrm{minc}}(r_1, r_2)$ is defined by analogy with (5.17).

Then we denote a function that normalizes a distance and returns values in the range [0,1] as:

$$B(d,t) = \frac{d}{t} \tag{5.18}$$

where t is a threshold value that specifies the neighbourhood of a label. In our implementation, we define the neighbourhood of a label with the parameter T_{d}, which is the minimum permissible distance. The parameter T_{d} is the threshold distance between the bounds of two labels.

For every two labels $\eta, \mu \in L$ we define a proximity function (*part 1*) as:

$$A_1(\eta,\mu) = \begin{cases} B(d_{\min}(\eta,\mu), T_{\mathrm{d}}), & d_{\min}(\eta,\mu) < T_{\mathrm{d}} \\ 1, & d_{\min}(\eta,\mu) \geq T_{\mathrm{d}}. \end{cases} \tag{5.19}$$

The function $A_1 : L \times L \to \mathbb{R}$ was designed to yield a value of 0.0 for a case when the labels η, μ touch each other and 1.0 if they are too far to raise any ambiguities.

Next, on the analogy of the function (5.19) we define the second function that measures the degree of alignment between two adjacent labels as:

$$A_2(\eta,\mu) = \begin{cases} B(d_{\mathrm{minc}}(\eta,\mu), T_{\mathrm{dc}}), & d_{\mathrm{minc}}(\eta,\mu) < T_{\mathrm{dc}} \\ 1, & d_{\mathrm{minc}}(\eta,\mu) \geq T_{\mathrm{dc}}. \end{cases} \tag{5.20}$$

where $d_{\mathrm{minc}}(\eta,\mu)$ is defined as:

$$d_{\mathrm{minc}}(\eta,\mu) = \begin{cases} 0, & \mathit{if}\ \eta\ \mathit{and}\ \mu\ \mathit{overlap} \\ \min\{d^x_{\mathrm{minc}}(r^s_\eta, r^n_\mu), d^x_{\mathrm{minc}}(r^n_\eta, r^s_\mu), d^x_{\mathrm{minc}}(r^n_\eta, r^n_\mu), & \mathit{otherwise} \\ \quad d^y_{\mathrm{minc}}(r^s_\eta, r^n_\mu), d^y_{\mathrm{minc}}(r^n_\eta, r^s_\mu), d^y_{\mathrm{minc}}(r^n_\eta, r^n_\mu)\}\ . & \end{cases} \tag{5.21}$$

Function (5.21) returns a minimum value of the set of distances that represent the distances between x- or y- components of the centres of the rectangles $r^s_\eta, r^n_\eta, r^s_\mu, r^n_\mu$, which in couples can bring ambiguity between two labels η, μ. It can be seen, that the function A_2 returns 0.0 if the centrelines of two rectangles (*CL1* and *CL2*, Figure 5.3a) coincide, 1.0 if the distance between the centrelines are greater than or equal to T_{dc}.

Next, we can combine the functions A_1 and A_2 into a single formula for $F^{\mathrm{disamb}}_{i,j}$ as follows:

$$F^{\mathrm{disamb}}_{i,j} = \begin{cases} \prod\limits_{y \in \tilde{L}(l_{i,j})} \left(\gamma_1 A_1(l_{i,j}, y) + \gamma_2 A_2(l_{i,j}, y)\right), & y \in \tilde{L}(l_{i,j}) \\ 1, \quad \tilde{L}(l_{i,j}) = \{\} & \end{cases} \tag{5.22}$$

where A_1, A_2 are defined distance functions, γ_1, γ_2 are the weights and should sum up to 1, $\tilde{L}(l_{i,j})$ is the neighbourhood of the label $l_{i,j}$ defined by T_d as a set $\tilde{L}(\eta) = \{\mu \in L|\ d_{\min}(\eta, \mu) < T_\mathrm{d}\}$. The metric (5.22) has multiplication which means that we compute the total degree of disambiguation between the label $l_{i,j}$ and its neighbouring labels $\tilde{L}(l_{i,j})$.

5.3.4.5 Cluttering

The purpose of this sub-section is to define a metric that can measure cluttering of neighbouring labels within a specific radius, see R6, and R7. We use a technique that was proposed by Ebner et al. (2003). In their work, the authors use a force-based model to compute placements with good label distribution in a short amount of time. The force-based model consists of two components: *intersection-proportional* and *distance-related*. We are interested in the second one which depends on the distance between two (R6). The dependency from distance is defined through the forces which tend to grow super linearly with decreasing the distance. In the definition of our metric we use formulas from the mentioned work, but with some modifications. At first let us make some definitions.

We define the clutter function as $g = (g_x, g_y)$. Let $c_\eta = (x_\eta, y_\eta)$ be the centre point of the minimum bounding rectangle r_η^n (Section 5.3.4.4) of a label $\eta \in L$, then the x-component of the clutter function is:

$$g_x(\eta, \mu) = \frac{g_d(x_\eta - x_\mu)}{\|c_\eta, c_\mu\|} \tag{5.23}$$

where $\|c_\eta, c_\mu\|$ is the Euclidean distance between two points, the function $g_d \colon L \times L \to \mathbb{R}$ is defined as follows:

$$g_d(\eta, \mu) = \frac{\delta}{\max(\varepsilon, d_{\min}(\eta, \mu))^2} \tag{5.24}$$

where $d_{\min}(\eta, \mu)$ is defined in (5.16). We define the y-component g_y analogously.

Once we have these components we can determine the resultant clutter function as:

$$v(\eta, \mu) = \sqrt{(g_x(\eta, \mu))^2 + (g_y(\eta, \mu))^2}\ . \tag{5.25}$$

Using (5.25) we define the cluttering measure as:

$$F_{i,j}^{\mathrm{clut}} = \begin{cases} \prod\limits_{y \in \tilde{L}(l_{i,j})} \left(\dfrac{v_{\max} - v(l_{i,j}, y)}{v_{\max} - v_{\min}} \right), & y \in \tilde{L}(l_{i,j}) \\ 1, & \tilde{L}(l_{i,j}) = \{\} \end{cases} \tag{5.26}$$

where $l_{i,j}$ is an element of L and $v_{\max}, v_{\min}$ are the maximum and minimum values of the clutter function v for all neighbouring labels of a label $l_{i,j}$. This neighbourhood of a label $\eta \in L$ is defined as $\widetilde{L}(\eta) = \{\mu \in L | \left\| c_\eta, c_\mu \right\| < T_{\text{clut}}\}$. T_{clut} is a parameter defined by the user according to the task at hand.

Note that both metrics $F_{i,j}^{\text{disamb}}$ and $F_{i,j}^{\text{clut}}$ measure how close two adjacent labels are to each other, and they are defined by different threshold values $T_{\text{d}}, T_{\text{dc}}$ and T_{clut}. Basically, a value of T_{clut} is much greater then values T_{d} and T_{dc}.

5.3.4.6 Coastal Places

One of the important cartographic requirements in map lettering on small scales is the set of rules that describe the principles of naming shore and coastal places (R4, R5). Following these rules the cartographer must make a reasonable compromise to achieve high legibility of a map and the actual topography that is depicted. Therefore, we tried to determine a metric that takes into consideration the rules R4 and R5.

By analogy with Section 5.3.4.3 we defined the set of axis-aligned rectangles that bound the characters of the name of the jth point feature on the ith position as $R_{i,j} = \left\{r_{i,j,1}, \dots, r_{i,j,K_j}\right\}$. This time the position and size of the rectangles are not aligned to any grid. We also assume that we have a vector dataset that consists of polygons which present water bodies. Then, let us define the following metric:

$$F_{i,j}^{\text{coast}} = \begin{cases} W_{i,j}, & W_{i,j} \geq T_{\text{w}}, l_{i,j} \in L^{\text{c}} \\ \min(1 - W_{i,j}, T_w), & W_{i,j} < T_{\text{w}}, l_{i,j} \in L^{\text{c}} \\ 1 - W_{i,j}, & l_{i,j} \notin L^{\text{c}} \end{cases} \tag{5.27}$$

where $W_{i,j}$ is a metric that measures the percentage of a label's area that lies on the water surface. It has the following form:

$$W_{i,j} = \frac{\sum_{r \in R_{i,j}} \widetilde{W}(r)}{A(R_{i,j})} \tag{5.28}$$

The function $\widetilde{W}$ returns a percentage of water inside a certain rectangle, which is defined by the bounds of a character. The function A returns the area of all rectangles in $R_{i,j}$. The $L^{\text{c}} \subseteq L$ denotes a set of labels that letters point features near the shore. We also defined the threshold parameter T_{w} that divides all coastal labels denoted as L^{c} into two groups. The first group is defined by $W_{i,j} \geq T_{\text{w}}$ and reflects that the labels lie mostly on the water surface. The second group is defined by $W_{i,j} < T_{\text{w}}$ and presents labels that are placed partially on water and on land. Furthermore, the expression for that group is defined in such a way that assigns higher priorities to the labels that are on land rather than on water. Until now, we have skipped one important issue, which is how to deter-

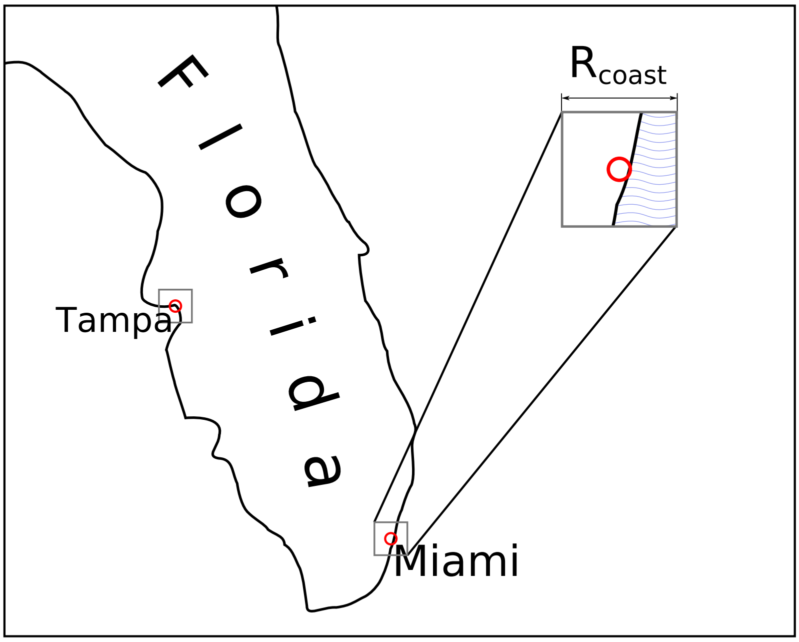


Figure 5.4. A scheme for computation of coastal places.

mine the set L^c. In our implementation, we construct a rectangle with the centre in the centre of a point feature and with the size equal to some parameter R_{coast} (Figure 5.4). Further we compute the area of intersection of the rectangle with all polygons in the given vector dataset. If the area of land in the rectangle is greater than a certain value T_{coast} we consider that the point feature belongs to L^c. The parameters $R_{\text{coast}}, T_{\text{coast}}$ are adjusted by the user. It is worth noting that on different scales the proposed approach returns different sets of L^c. This scale-dependent behavior does not contradict any cartographic requirement. Moreover, for a certain scale, it properly evaluates the relationship between a label and a coastline, for cases when the coastal position is not defined in a data source.

5.4 Experimental Results

We implemented the multi-criteria model within a framework for publishing spatial data to the web. This framework is called *MapSurfer.NET* and written in C#. We ran our experiments on a machine with an Intel® Core™ i5-2500 CPU @ 3.30 GHz running Windows 7 Professional x64 with 8GB installed memory. The runtime execution environment of our test application was .NET Framework 4.5 (x64).

In our implementation we used a couple of techniques that allow an increase in performance of label placement algorithms. Firstly, we made use of a quad tree data structure (Finkel and Bentley, 1974) to store labels and examine whether any label overlaps other characters or symbols on the map. Secondly, in a pre-processing step, we construct a conflict graph, whose nodes are all labels of the map, and whose edges indicate potential overlap with other labels (nodes). The graph-based approach is one of the most common methods in the field of interactive and dynamic labelling (Been et al., 2006; Mote, 2007).

In order to compare effectiveness and accuracy of our model we used three well-known heuristic search algorithms for solving PFLP as a mathematical programming problem. They are *greedy* (Yoeli, 1972; Christensen et al., 1995), *discrete gradient descent* (Christensen et al., 1995) and *simulated annealing* (Christensen et al., 1995; Zoraster, 1997; Edmondson et al., 1996) algorithms. In our experiments these algorithms are used to find a feasible near-optimal solution for label placement by treating the proposed model as the *objective function* to be optimized. In our implementation the greedy algorithm is restricted to the selection of the first candidate position for a point feature that can be placed on a map, i.e. the improvement of a final solution within the candidate positions of a point feature is not allowed. As an annealing schedule for the simulated annealing algorithm we chose a polynomial-time cooling schedule that was proposed by Aarts and van Laarhoven (1985). We should notice that the first two algorithms perform the search on a local basis only. The major weakness of these algorithms is that they can be trapped in local minima of the objective function. In contrast, the stochastic nature of simulated annealing helps escape the local minima. Furthermore, simulated annealing is very simple in implementation and returns nearly optimal solutions with relatively good performance (Christensen et al., 1995).

We performed our experiments on a dataset that represents geospatial data granted by the *OpenStreetMap* project that is one of the most promising crowd sourced projects. The test dataset represented the northern part of Denmark. From the dataset we extracted all settlements and divided them into 4 groups. To each group we assigned a different font size and image for point features which reflect the population and administrative status of a place.

Below we discuss the settings and parameters used in the tests. The parameters of the simulated annealing algorithm are: the maximum number of temperature stages is $500N$ where N is the number of point features. At each temperature a maximum of $5N$ candidate positions are repositioned. The maximum number of iterations without any improvement in the solution is $400N$. The initial value of temperature is 100.0 and the minimum value is 0.000000001. The parameters settings of the model are: the weights in equation (5.6) $\alpha_1 = 0.6$ and $\alpha_2 = 0.4$. We set $\alpha_1 > \alpha_2$ by pursuing the primary goal of placing as many labels as possible. The parameters of the metrics are: $\gamma_1 = 0.7, \gamma_2 =$

$0.3, T_{\text{d}} = 8$ $T_{\text{dc}} = 5$, $T_{\text{clut}} = 30$, $R_{\text{coast}} = 12, T_{\text{coast}} = 0.2$ which are measured in map units (pixels in our tests), $\delta = 1, \varepsilon = 0.5$ and $T_{\text{w}} = 0.9$. A raster of the map background for the feature overlapping metric is based on the same OSM dataset (coastline, water bodies and roads). For image segmentation we used the *octree* colour quantization algorithm (Gervautz and Purgathofer 1988) with the number of colours equal to 256.

Five variations of model weights $\beta_j,\ j = 1, \dots, 6$ are presented in Table 5.3 that examine the impact of each of them on final map lettering and on the performance of a PFLP algorithm.

In Figure 5.5 we present the results of Test №1 for the greedy and gradient descent algorithms respectively. In this test we used only two metrics common in the literature, such as positioning around a point feature and feature hierarchy. From these figures it is clear that the resulting map does not provide a high level of functionality due to lettering that partially hides some important and relevant geographic features like roads (Figure 5.5a, ref. 2) and bays (Figure 5.5b, ref. 1). Moreover, we can see some distinct ambiguities between the names and features they label (see ref. 1 in Figure 5.5a, 5.5b).

In the second test we added the feature-overlapping metric. Figure 5.6a depicts the resulting map by using three quality metrics and uses the simulated annealing algorithm. As expected, we were able to considerably increase the number of labelled point features to 73. It can be seen from the Figure 5.6a that the model strives to move labels towards homogeneous parts of the map. The names of the coastal places are placed on the water surface. All but a few label placements seem more or less acceptable. For example, the name of the town Nykøbing Mors is placed partially on water and land by totally concealing a rather big island and a bay (Figure 5.6a, ref. 1). It can be explained by the fact that the feature-overlapping metric considers the overlap of water or roads equivalently.

To help the model to find an unambiguous solution in this case we made use of the metric $F_{i,j}^{\text{coast}}$. Figure 5.6b shows nearly the same positioning of the names as in Figure 5.6a, but now the name of the coastal town Nykøbing Mors is on the land (Figure 5.6b, ref. 1).

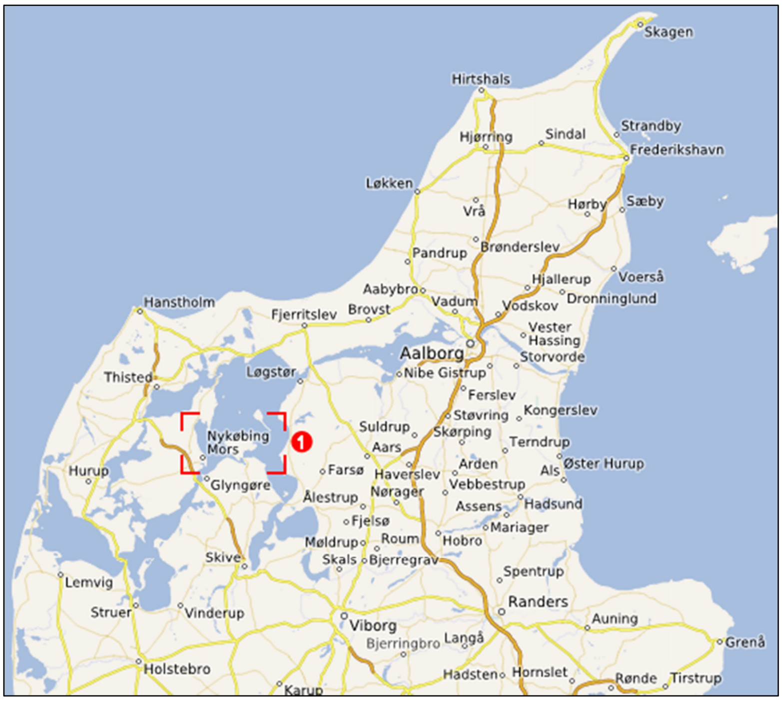
Table 5.3. The results of the PFLP algorithms with different parameters of the model.

№	N	β_1	β_2	β_3	β_4	β_5	β_6	Greedy				Gradient Descent				Simulated Annealing			
								N_l	%	$Q(x)$	$t_{\text{cpu}}[s]$	N_l	%	$Q(x)$	$t_{\text{cpu}}[s]$	N_l	%	$Q(x)$	$t_{\text{cpu}}[s]$
1	82	0.6	0.4	0	0	0	0	59	71.95	0.617	0.01908	66	80.5	0.657	0.02553	73	89.0	0.706	1.0298
2	82	0.3	0.2	0.5	0	0	0	59	71.95	0.673	0.08279	66	80.5	0.717	0.09524	73	89.0	0.784	1.2970
3	82	0.2	0.1	0.3	0	0	0.4	59	71.95	0.712	0.39249	66	80.5	0.761	0.42134	74	90.2	0.839	1.4876
4	82	0.2	0.1	0	0	0	0.7	59	71.95	0.729	0.34698	66	80.5	0.779	0.35373	74	90.2	0.858	1.5752
5	82	0.2	0.1	0.3	0.1	0.05	0.25	59	71.95	0.710	0.50949	66	80.5	0.760	0.51101	74	90.2	0.834	3.5692

(a) Greedy

(b) Discrete Gradient Descent

Figure 5.5. The resulting maps of Test №1.



(a) Test №2

(b)Test №3

Figure 5.6. The maps resulting from the simulated annealing algorithm.

(a) Test №4

(b) Test №5

Figure 5.7. The maps resulting from the simulated annealing algorithm.

In our next experiment we carried out by taking into account only the following metrics $F_{i,j}^{\text{prior}}$, $F_{i,j}^{\text{pos}}$, $F_{i,j}^{\text{coast}}$. In this test (Figure 5.7a) we observe more labels that conceal roads. For example, the names of settlements Hostebro, Randers, and Auning (Figure 5.7a, ref. 2) entirely overlap some parts of rather important roads that form a national route network. Hence, we can draw the conclusion that even a simple variant of the feature-overlapping metric performs surprisingly well.

From Figures 5.5a, 5.5b, 5.6a, 5.6b, 5.7a it is obvious that there is a high degree of ambiguity between labels and their point features and that the labels are visually cluttered. A demonstrative example is a group of villages that consist of Møldrup, Roum, Bjerregrav, and Skals (Figure 5.6b, ref. 2, Figure 5.7a ref. 1).

The final test we ran with all six metrics. The map in Figure 5.7b illustrates a label assignment that is less cluttered than the previous ones (Figure 5.7b, ref. 1). That group of villages is now labelled clearer and without ambiguities (Figure 5.8).

It is evident from the tabular data (Table 5.3) that the simulated annealing algorithm predominated over other algorithms in the quality of the label assignment, while the greedy heuristic outperforms other algorithms by always returning the worst solution for the PFLP problem. It is also clear that computational resources required for each PFLP algorithm and set of metrics vary greatly. In order to determine how strong the impact of different metrics is upon performance, we calculated a score as the running time of Test №5 divided by the running time of Test №1. For the tested PFLP algorithms this score is 26.70, 20.01 and 3.46 respectively. Comparing the scores, we can conclude that a substantial amount of computation time is spent on pre-processing for different metrics.

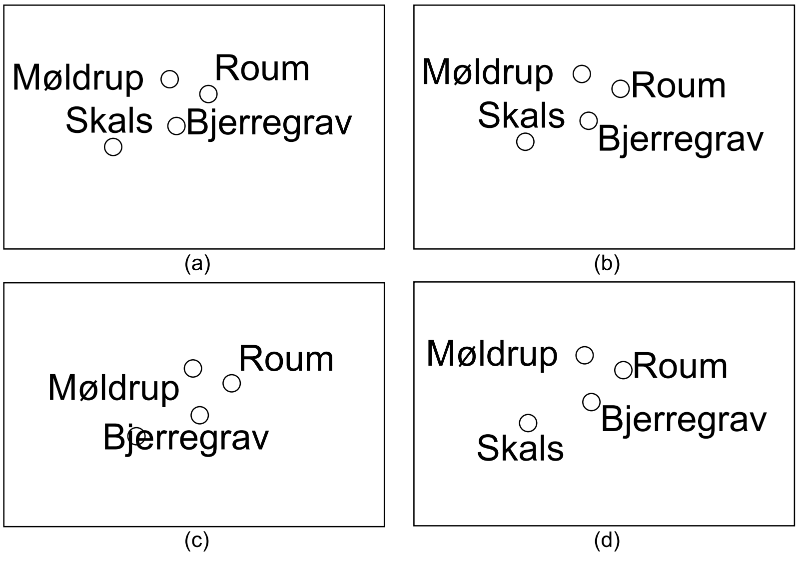


Figure 5.8. A sketchy representation of ambiguity for a group of villages: (a) Figure 5.6a; (b) Figure 5.6b; (c) Figure 5.7a; (d) Figure 5.7b.

5.5 Conclusions

In this paper we presented a multi-criteria optimization model for automated label placement for point features in cartography. Our complete model of a quality evaluation function for the PFLP problem satisfies almost all cartographic requirements for point features more than any other previous model given in the literature. Producing unambiguously labelled maps has been traditionally recognized as being the most important aim of the whole lettering process (Brewer, 2005; Imhof, 1962/1975; Wood, 2000). To our knowledge this is the first attempt to address, among other things, the quantification of ambiguous label-feature relations. The proposed model is highly adjustable and provides the human cartographer a handy tool to make an appropriate label placement according to his preferences. It also conceptually opens the possibility to automate the preceding stages of cartographic lettering (conceptual toponyms transformation/filtering, task-based toponym filtering, font selection), which have previously been neglected in the re-

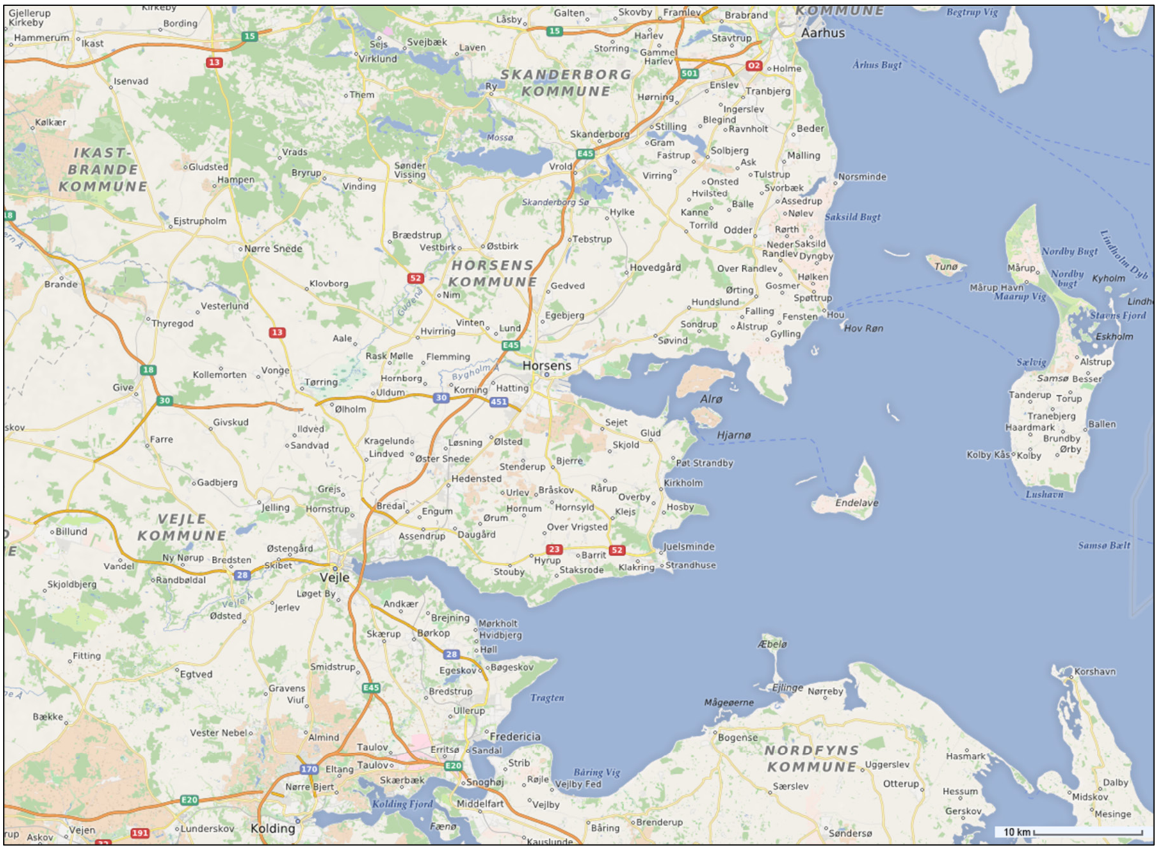


Figure 5.9. A sample map involving area, line and point features labelling using the presented model. Projection: spherical Mercator. Data source: © OpenStreetMap contributors (data licensed under ODbL). Available at http://openmapsurfer.uni-hd.de.

search. The experiments argue that the model together with an appropriate mathematical optimization algorithm for PFLP, which is able to find a good approximation to the global optimum, produces visually *plausible* lettering with high cartographic quality and is capable of considerably enhancing the functionality of the map.

It is worth noting that the presented model can also be used for labelling other feature types (lines, areas). The difference consists in modifying a part of the algorithm that generates and evaluates the candidate positions and applying appropriate models (for example, line-features: Barrault and Lecordix, 1995; Edmondson et al., 1996; area-features: van Roessel, 1989; Barrault, 2001). Furthermore, the metric $F_{i,j}^{\text{coast}}$ should be omitted and $F_{i,j}^{\text{disamb}}$ should be slightly altered. Figure 5.9 depicts a sample map that contains labels of area, line, and point features that were labelled using an extended version of our model adjusted for lines and areas. For the sake of comparison, we have provided, in Figure 5.10, a map of the same region which shows the labelling performed by the greedy algorithm of the standard OSM rendering done by the *Mapnik* toolkit without any quality evaluation. The difference in labelling quality and toponym density should be apparent.

For future work, we plan to improve the feature-overlapping metric (Section 5.3.4.3). In spite of the fact that this metric showed its efficiency in our tests, it leads to an ambiguous behavior in some cases. More exactly, it does not take into account the spatial distribution of the mapped features under a name. While topological and semantic relationships probably need ancillary data, widening the approach to include curved labels seems straightforward.

We implemented the model on top of the platform *MapSurfer.NET* for publishing spatial data to the web. We prepared and published a web map that is based on the real global dataset provided by the *OpenStreetMap* project. It is available online on the *OpenMapSurfer* (2013) web page (OSM Roads layer). The name placement on small scales (zoom levels 2–12) is done using the multi-criteria model devised in this paper.

Acknowledgements

The authors would like to express their particular appreciation to the anonymous reviewers for the constructive and helpful comments and suggestions. We acknowledge colleagues at our institute for their valuable feedback. We also wish to personally thank Alexander Zipf for supervising this research and the University of Heidelberg for providing the hardware and making the results of the research available through the web map service. Furthermore, Maxim Rylov gratefully acknowledges DAAD (Deutscher Akademischer Austauschdienst, German Academic Exchange Service) for awarding the DAAD-Scholarship to pursue his doctoral degree.

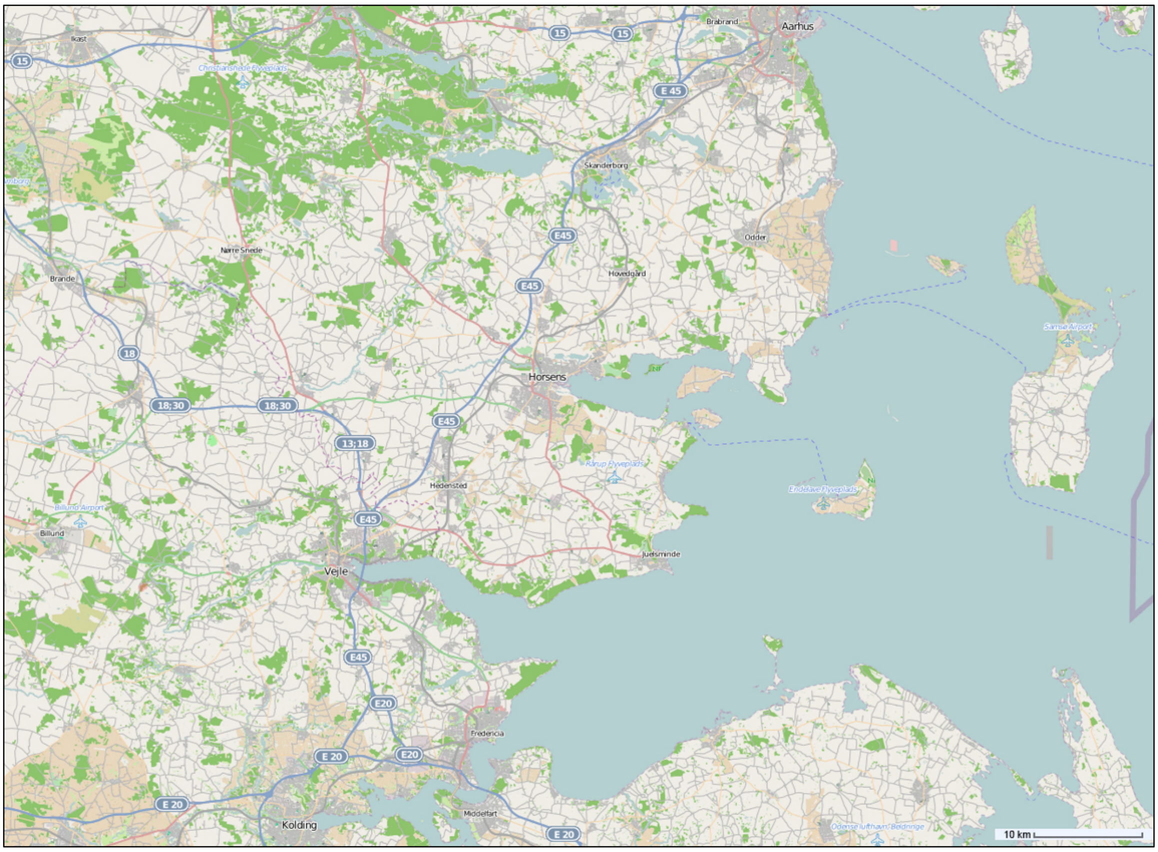


Figure 5.10. A sample map with point-feature labelling which was taken from http://www.openstreetmap.org. Projection: spherical Mercator, Data source: © OpenStreetMap contributors (data licensed under ODbL; cartography licensed under CC BY-SA).

References

Aarts, E.H.L., and P.J.M. van Laarhoven. 1985. "A New Polynomial Time Cooling Schedule." In Proc. *IEEE International Conf. on Computer Aided Design*, Santa Clara, 206-228.

Ahn, J., and H. Freeman 1984. "A Program for Automatic Name Placement." *Cartographica* 21(2-3): 101-109. doi: 10.3138/0646-Q262-6636-3681

Anson, R.W. (Ed.) 1988. Basic Cartography for Students and Technicians. vol. 2. London: ICA/Elsevier.

Bae, W.D, S. Alkobaisi, S. Narayanappa, P. Vojtechovský, and K. Bae. 2011. "Optimizing Map Labeling of Point Features Based on an Onion Peeling Approach." *Journal of Spatial Information Science* 2: 3-28. doi: 10.5311/JOSIS.2011.2.4

Barrault, M. 2001. "A Methodology for Placement and Evaluation of Area Map Labels." *Computers, Environment and Urban Systems* 25(1): 33-52. doi: 10.1016/S0198-9715(00)00039-9

Barrault, M., and F. Lecordix. 1995. "An Automated System for Linear Feature Name Placement which Complies with Cartographic Quality Criteria." In *Proc. AutoCarto 12:* 321-330. ACSM/ASPRS, Bethesda.

Been, K., E. Daiches, and C. Yap. 2006. "Dynamic Map Labeling." *IEEE Transactions on Visualization and Computer Graphics* 12(5): 773-780. doi: 10.1109/TVCG.2006.136

Bertin, J. 1983. Semiology of Graphics: diagrams, networks, maps. University of Wisconsin Press.

Brewer, C.A. 2005. Designing Better Maps: A Guide for GIS Users, ESRI Press, Redlands CA.

Butzler, S.J., C.A. Brewer, and W.J. Stroh. 2011. "Establishing Classification and Hierarchy in Populated Place Labeling for Multiscale Mapping for The National Map." *Cartography and Geographic Information Science* 38(2): 100-109. doi: 10.1559/15230406382100

Christensen, J., J. Marks, and S.M. Shieber. 1995. "An Empirical Study of Algorithms for Point-Feature Label Placement." *ACM Transactions on Graphics* 14(3): 203-232. doi: 10.1145/212332.212334

Cook, A.C., and C.B. Jones. 1990. "A Prolog Rule-Based System for Cartographic Name Placement." *Computer Graphics Forum* 9(2): 109-126. doi: 10.1111/j.1467-8659.1990.tb00384.x

Cromley, R.G. 1985. "An LP Relaxation Procedure for Annotating Point Features Using Interactive Graphics." In *Proceedings of the AutoCarto 7,* 127-132.

Doerschler, J.S., and H. Freeman. 1992. "A Rule-Based System for Dense-Map Name Placement." *Communications of the Association of Computing Machinery* 35(1): 68-79. doi: 10.1145/129617.129620

Ebner, D., G.W. Klau, and R. Weiskircher. 2003. "Force-Based Label Number Maximization." Technical Report TR-186-1-03-02, Vienna University of Technology.

Edmondson, S., J. Christensen, J. Marks, and S. Shieber. 1996. “A General Cartographic Labeling Algorithm.” *Cartographica* 33(4): 13-23. doi: 10.3138/U3N2-6363-130N-H870

Finkel, R., and J.L. Bentley. 1974. “Quad Trees: A Data Structure for Retrieval on Composite Keys.” *Acta Informatica* 4(1): 1-9. doi: 10.1007/BF00288933

Formann, M., and F. Wagner. 1991. “A Packing Problem with Applications to Lettering of Maps.” In *Proceedings of the 7th Annual Symposium on Computational Geometry*, 281-288.

Freeman, H., and J. Ahn. 1984. “AUTONAP – An Expert System for Automatic Map Name Placement.” In *Proc. 1st International Symp. on Spatial Data Handling*, 544-569.

Freeman, H., and J. Ahn. 1987. “On the Problem of Placing Names in a Geographic Map.” *International Journal of Pattern Recognition and Artificial Intelligence* 1(1): 121-140. doi: 10.1142/S0218001487000096

Gervautz, M., and W. Purgathofer. 1988. “A Simple Method for Color Quantization: Octree Quantization.” *New Trends in Computer Graphics*, ed. N. Magnenat-Thalmann and D. Thalmann. Springer Berlin Heidelberg. 219-231. doi: 10.1007/978-3-642-83492-9_20

Gomes, S.P, G.M. Ribeiro, and L.A.N. Lorena. 2013. “Dispersion for the Point-Feature Cartographic Label Placement Problem.” *Expert Systems with Applications* 40(15): 5878-5883. doi: 10.1016/j.eswa.2013.04.035

Goodchild, M.F. 2007. “Citizens as Sensors: The World of Volunteered Geography.” *GeoJournal* 69: 211-221. doi: 10.1007/s10708-007-9111-y

Haklay, M., and P. Weber. 2008. “OpenStreetMap – User Generated Street Map”, IEEE Pervasive Computing 7(4): 12-18. doi: 10.1109/MPRV.2008.80

Haralick, M. and Shapiro, L. G. 1985. “Image segmentation techniques.” *Computer Vision Graphics and Image Processing* 29: 100-132. doi: 10.1016/S0734-189X(85)90153-7

Harrie, L., H. Stigmar, T. Koivula, and L. Lehto. 2004. “An Algorithm for Icon Placement on a Real-Time Map.” In *Developments in spatial data handling*, ed. P. Fisher. 493-507. Berlin: Springer.

Hirsch, S.A. 1982. “An Algorithm for Automatic Name Placement around Point Data.” *American Cartographer* 9(1): 5-17. doi: 10.1559/152304082783948367

Huffman, F.T., and R.G. Cromley. 2002. “An Automated Multi-Criteria Cartographic Aid for Point Annotation.” *Cartographic Journal* 39(1): 51-64. doi: 10.1179/caj.2002.39.1.51

Imhof, E. 1962. “Die Anordnung der Namen in der Karte.” In *International Yearbook of Cartography*, 93-129, Bonn-Bad Godesberg: Kirschbaum.

Imhof, E. 1972. Thematische Kartographie. Berlin, Walter de Gruyter.

Imhof, E. 1975. “Positioning names on maps.” *American Cartographer* 2(2): 128-144. doi: 10.1559/152304075784313304

Jones, C. B. 1989. “Cartographic Name Placement with Prolog.” *IEEE Computer Graphics and Applications* 9(5): 36-47. doi: 10.1109/38.35536

Kato, T., and H. Imai. 1988. “The NP-completeness of the Character Placement Problem of 2 or 3 Degrees of Freedom.” Record of Joint Conference of Electrical and Electronic Engineers in Kyushu, 11-18. [In Japanese.]

Keates, J.S. 1973. Cartographic Design and Production. London: Longman.

Kern, J.R., and C.A. Brewer. 2008. “Automation and the Map Label Placement Problem: A Comparison of Two GIS Implementations of Label Placement.” *Cartographic Perspectives* 60, 22-45.

Kirkpatrick, S., Jr.C.D. Gelatt, and M.P. Vecchi. 1983. “Optimization by Simulated Annealing.” *Science* 220: 671-680. doi: 10.1126/science.220.4598.671

Klau, G.W., and P. Mutzel. 2003. “Optimal labeling of point features in rectangular labeling models.” *Mathematical Programming* 94(2-3):435-458. doi: 10.1007/s10107-002-0327-9

Luboschik, M., H. Schumann, and H. Cords. 2008. “Particle-Based Labeling: Fast Point-Feature Labeling without Obscuring Other Visual Features.” *IEEE Transactions on Visualization and Computer Graphics* 14(6): 1237-1244. doi: 10.1109/TVCG.2008.152

Marks, J., and S. Shieber. 1991. “The computational complexity of cartographic label placement.” Technical Report TR-05-91, Harvard University, March.

Morrison, J.L. 1980. “Computer Technology and Cartographic Change.” In *The Computer in Contemporary Cartography*, ed. D.R.F. Taylor, 5-23. New York: Wiley.

Mote, K. 2007. “Fast Point-Feature Label Placement for Dynamic Visualizations.” *Information Visualization* 6(4): 249-260. doi: 10.1057/PALGRAVE.IVS.9500163

OpenMapSurfer, 2013. (a web map service provided by University of Heidelberg; accessed May 14, 2013). Available at http://openmapsurfer.uni-hd.de.

Ramm, F., J. Topf, and S. Chilton. 2010. *OpenStreetMap: Using and Enhancing the Free Map of the World*. Cambridge: UIT Cambridge.

Raposo, P., C. Brewer, and L. Stanislawski. 2013. “Label and Attribute-Based Topographic Point Thinning.” In *Proc. ICA Workshop on Generalisation and Multiple Representation*, August 25-30, Dresden, Germany.

Regnauld, N, S. Lessware, C. Wesson, and P. Martin. 2013. “Deriving Products from a Multiple Resolution Database using Automated Generalisation at Ordnance Survey.” In Proc. 26th International Cartographic Conference, August 25-30, Dresden, Germany.

Revell, P., N. Regnauld, and G. Bulbrooke. 2011. “OS Vectormap District: Automated Generalisation, Text Placement and Conflation in Support of Making Public Data Public.” In Proc. 25th International Cartographic Conference, July 3-8, Paris, France.

Robinson, A.H., J.L. Morrison, P.C. Muehrcke, A.J. Kimerling, and S.C. Guptil 1995. *Elements of Cartography*, 6th ed. New York: Wiley.

Schreyer, M., and G.R. Raidl. 2002. “Letting Ants Labeling Point Features.” In Proc. IEEE Congress on Evolutionary Computation (CEC '02) 2: 1564-1569.

Stadler, G., T. Steiner, and J. Beiglbock. 2006. “A Practical Map Labeling Algorithm Utilizing Morphological Image Processing and Force-Directed Methods.” *Cartography and Geographic Information Science* 33(3): 207-215. doi: 10.1559/152304006779077327

Strijk, T.W., and M.J. van Kreveld. 2002. “Practical Extensions of Point Labeling in the Slider Model”, *GeoInformatica* 6(2): 181-197. doi: 10.1023/A:1015202410664

van Altena, V., R. Nijhuis, M. Post, B. Bruns, and J. Stoter. 2013. “Automated generalisation in production at Kadaster NL.” In *Proc. 26*th *International Cartographic Conference*, August 25-30, Dresden, Germany.

van Dijk, S., D. Thierens, and M. de Berg. 2004. “On the Design and Analysis of Competent Selecto-Recombinative GAs.” *Evolutionary Computation* 12(2): 243-267. doi: 10.1162/106365604323142842

van Dijk, S., M.J. van Kreveld, T.W. Strijk, and A. Wolff. 2002. “Towards an Evaluation of Quality for Names Placement Methods.” *International Journal of Geographical Information Science* 16(7): 641-661. doi: 10.1080/13658810210138742

van Kreveld, M.J., T.W. Strijk, and A. Wolff. 1999. “Point Labeling with Sliding Labels.” *Computational Geometry: Theory and Applications* 13: 21-47. doi: 10.1016/S0925-7721(99)00005-X

van Roessel, W. 1989. “An Algorithm for Locating Candidate Labeling Boxes Within a Polygon.” *American Cartographer* 16(3): 201-209. doi: 10.1559/152304089783814034

Verner, O.V., R.L. Wainwright, and D. Schoenefeld. 1997. "Placing Text Labels on Maps and Diagrams using Genetic Algorithms with Masking." *INFORMS Journal on Computing* 9(3): 266-275. doi: 10.1287/ijoc.9.3.266

Wolff, A., and T. Strijk, 2009. "A map labeling bibliography." Available at http://i11www.iti.uni-karlsruhe.de/~awolff/map-labeling/bibliography/.

Wood, C.H. 2000. "Descriptive and Illustrated Guide for Type Placement on Small Scale Maps." *Cartographic Journal* 37(1): 5-18. doi: 10.1179/000870400787320851

Wu, C.V. and B.P. Buttenfield. 1991. "Reconsidering Rules for Point-Feature Name Placement." *Cartographica* 28(1): 10-27. doi: 10.3138/Y121-5621-8888-5306

Yamamoto, M., G. Camara, and L.A.N. Lorena. 2002. "Tabu Search Heuristic for Point-Feature Cartographic Label Placement." *Geoinformatica* 6(1): 77-90. doi: 10.1023/A:1013720231747

Zoraster, S. 1986. "Integer Programming Applied to the Map Label Placement Problem." *Cartographica* 23(3): 16-27. doi: 10.3138/9258-63QL-3988-110H

Zoraster, S. 1990. "The Solution of Large 0-1 Integer Programming Problems Encountered in Automated Cartography." *Operations Research* 38(5): 752-759. doi: 10.1287/opre.38.5.752

Zoraster, S. 1997. "Practical Results using Simulated Annealing for Point Feature Label Placement." *Cartography and Geographic Information Science* 24(4): 228-238. doi: 10.1559/152304097782439259

6 A Practical Algorithm for the External Annotation of Area Features

Authors	Maxim A. Rylov and Andreas W. Reimer
Journal	*The Cartographic Journal*
History	Published (online): 16 September 2014 Accepted: 25 April 2014 Revised: 24 March 2014 Submitted: 2 October 2013
Reference	Rylov, M.A., and A.W. Reimer. 2014. “A practical algorithm for the external annotation of area features.” *Cartographic Journal*. doi: 10.1179/1743277414Y.0000000091

6 A Practical Algorithm for the External Annotation of Area Features

Abstract

One of the subtasks of automated map labelling that has received little attention so far is the labelling of areas. Geographic areas often are represented by concave polygons which pose severe limitations on straightforward solutions due to their great variety of shape, a fact worsened by the lack of measures for quantifying feature-label relationships. We introduce a novel and efficient algorithm for labelling area features externally, i.e. outside their polygonal boundary. Two main contributions are presented in the following. First, it is a highly optimized algorithm of generating candidate placements utilizing algorithms from the field of computational geometry. Second, we describe a measure for scoring label positions. Both solutions based on a series of well-established cartographic precepts about name positioning in the case of semantic enclaves such as islands or lakes. The results of our experiments show that our algorithm can efficiently place labels with a quality that is close to the quality of traditional cartographic products made by human cartographers.

Keywords: automated cartography, automated label placement, area lettering, computational geometry, GIS mapping

6.1 Introduction

Being one of the key factors of the cartographic representation (Imhof, 1975; Robinson et al., 1995), name positioning is one of the most difficult and time-consuming tasks of map design process. Map functionality and legibility are highly affected by the quality of name positioning on maps. The first attempt to automate map lettering was made in 1972 by Yoeli. In spite of numerous research attempts during the last four decades (see extensive bibliography of papers on this topic maintained by Wolff and Strijk, 2009) to automate the lettering in map-making (Freeman, 2005), the problem is still vital and remains particularly challenging.

This paper addresses the problem of automated labelling of area features outside their boundary (see Figure 6.1) on small and medium scales. Among those are natural features such as small islands, lakes, valleys, canyons or urban areas. Such a problem arises in cartography basically in two cases:

- The area is too small to place the label entirely inside.

- Any internal label overlaps other labels or completely conceals other important geographic features of the map.

On a more conceptual level, we find circumstances leading to the desire for external labelling of polygons repeat themselves. What lakes and islands have in common for the case of maps is that they are, conceptually, semantic enclaves. Such noteworthy areas within a semantic sea of 'the other/the rest' can be encountered in other graphical domains, too. Examples where external polygon labelling might be beneficial include 2D outlier visualization and displaying classification or clustering results. Especially where the semantic enclaves are surrounded by an entity with much lower graphical density, placing text outside the area feature to be annotated is an attractive and tried technique. Positioning names in regions with lower graphic complexity (Castner and Eastman, 1985) is also tempting in the sense of higher type legibility (Phillips et al., 1977) and legibility of the map as a whole.

To the best of our knowledge, there are no extant automated methods for this task that can be found in the literature. Nevertheless, tools that are able to provide comparable functionality do exist. Among them are ESRI's *Maplex Label Engine* (2009) and

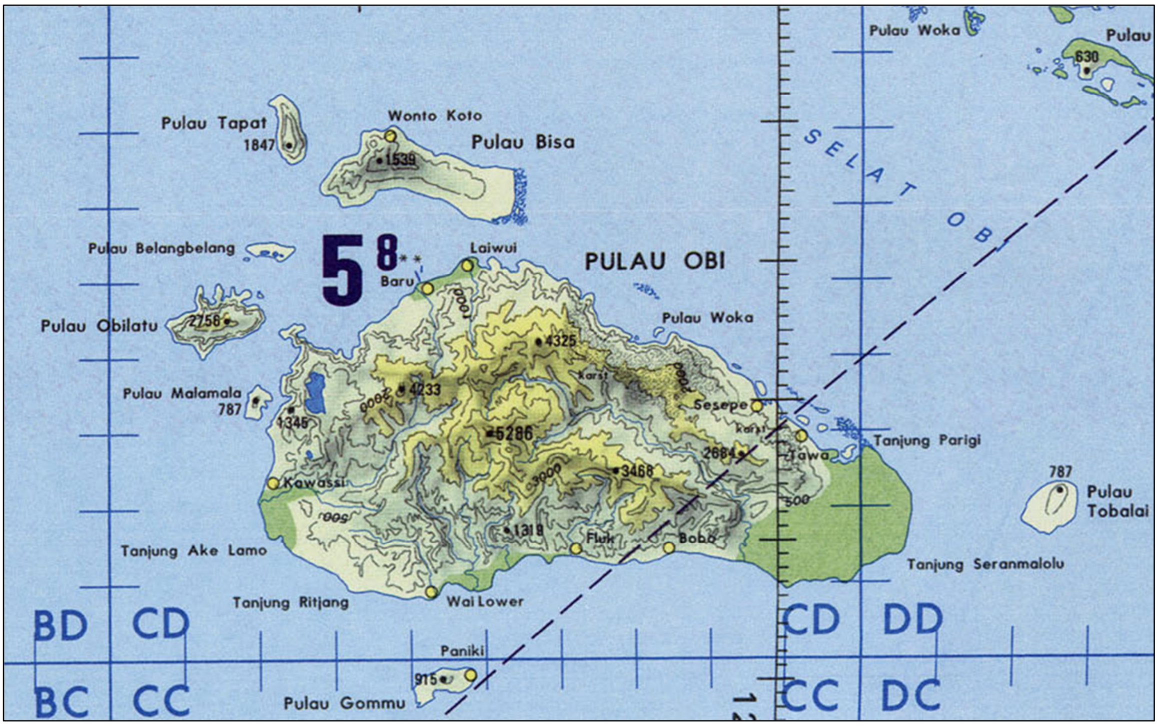


Figure 6.1. An example of manual lettering of Indonesian islands. Note how the multiple use of a full black and other dark print colours on the island's interior (hill shading, height information, isolines, etc.) forces the 'Pulau Obi' and 'Pulau Bisa' labels to the outside of their polygons Source: Army Topographic Support Establishment (1997).

Table 6.1. Availability of a method for external labelling of areal features in the existing toolkits and frameworks.

Toolkit	Maplex (ArcGIS)	Label-EZ (MapText)	MapInfo	PAL (QGIS)	MapServer	GeoServer	Mapnik
Availability	X	X	-	-	-	-	-

MapText's *Label-EZ* (2014). However, it can be seen from Table 6.1 that most other popular toolkits do not have a method to place the labels outside the areal feature. Table 6.1 clearly indicates that such functionality is implemented only in proprietary software packages and its description is not publicly available. Being provided in sufficient detail, the proposed algorithm can be easily reproduced. It can potentially extend capabilities of any label placement toolkit and thus, can partially fill a gap between open source and commercial packages.

We start by introducing automated label placement and briefly reviewing previous work in that field (Section 6.2). Further, we describe our approach in detail (Section 6.3) including a discussion of the cartographic requirements (Section 6.3.1). Then we present a robust and highly optimized *plane sweep* algorithm (Section 6.3.2) for generating *candidate positions*. In Section 6.3.3 we consider an evaluation function that measures ambiguous relationship between a label and the area feature it tags. In Section 6.4 we describe the results of our extensive empirical tests which we carried out on the dataset based on *Volunteered Geographic Information* (VGI; Goodchild, 2007), namely on the data of *OpenStreetMap* (OSM; Haklay and Weber, 2008; Ramm et al., 2010) project. The experimental results show that the proposed approach is able to produce plausible and functional label placement that satisfies cartographic criteria for the feature types we are interested in. We also provide a visual comparison between the name positioning generated with our method and the type placement produced by Maplex and Label-EZ (Section 6.4.4). Finally, we give some concluding remarks of the presented approach (Section 6.5).

6.2 Related Work

Automated label placement problem plays an important and critical role in several disciplines such as cartography (Yoeli, 1972; Monmonier, 1982), Geographic Information Systems (GIS; Freeman, 1991), and chart and graph drawing (Battista et al., 1994; Kakoulis and Tollis, 2003). Being one of the most difficult and complex problems of the mentioned disciplines, the label placement problem is generally split up into smaller and simpler independent sub-problems (or subtasks). These subtasks are (see Figure 6.2):

1. *Candidate position generation;*
2. *Position evaluation;*
3. *Position selection.*

The first subtask consists of generating a set of label candidates for each map feature by taking into account its type (point, line and polygon), shape as well as well-defined cartographic principles (Imhof, 1975; Wood, 2000). There were various research works which devised different solutions to tackle this subtask. Point-feature label placement (PFLP) is the problem of assigning text to point features objects (settlements, mountain peaks, points of interest, etc.). The PFLP is known to be an *NP-hard* problem (Kato and Imai, 1988; Marks and Shieber, 1991; Formann and Wagner, 1991). For point-like objects two different labelling models are differentiated, namely, *fixed position* (Yoeli, 1972; Hirsch, 1982) and *sliding label* models (van Kreveld et al., 1999; Strijk and van Kreveld, 1999; Klau and Mutzel, 2000). The task of tagging linear features (e.g., roads, rivers, boundaries) requires special and more sophisticated methods of generating potential positions. The research for labelling these feature types was conducted by Barrault and Lecordix (1995), Edmondson et al. (1996), Chirié (2000) and Wolff et al. (2001). Areal features are usually labeled first due to lesser placement flexibility. This issue is the most challenging due to the possible complexity of shapes. Nevertheless, some practical solutions for placing names inside polygons were suggested by Carstensen (1987), van Roessel (1989), Pinto and Freeman (1996), Barrault (2001), Dörschlag et al. (2003).

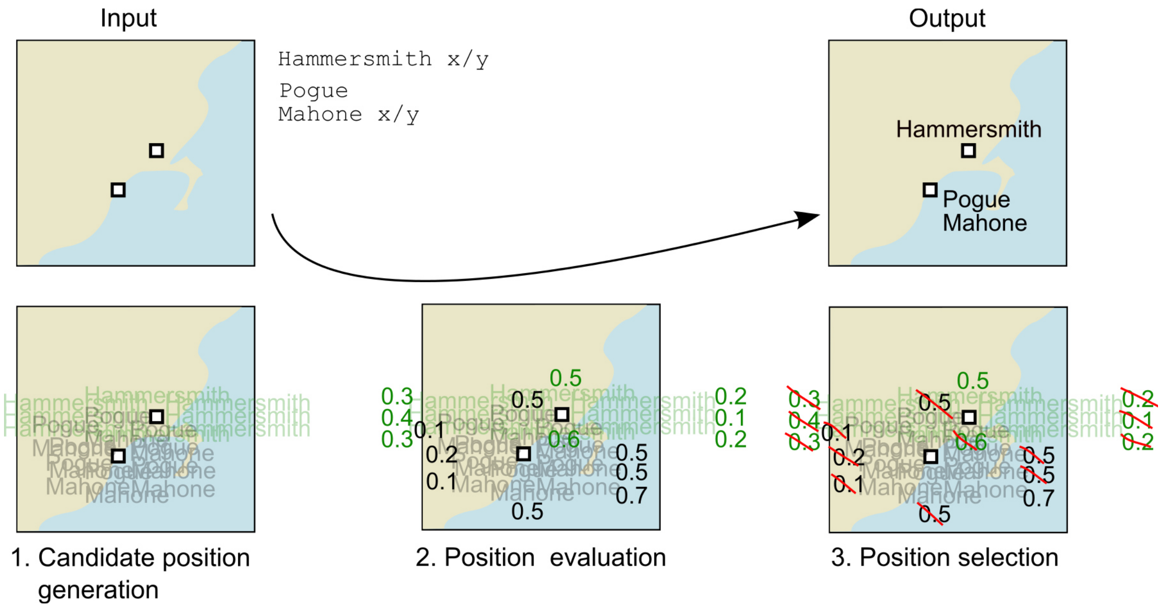


Figure 6.2. The subtasks of automated label placement.

The second subtask (*Position evaluation*) is the process of evaluating the quality of possible positions (subtask 1) by measuring how well a label is positioned with respect to the feature it tags and to the rest of the map content (van Dijk et al., 2002; Hong et al., 2005). Manifold metrics of evaluating quality of candidate positions for different feature types can be found in the works of Barrault and Lecordix (1995), Edmondson (1996), Chirié (2000), Barrault (2001), Rylov and Reimer (2014).

The third subtask (*Position selection*) lies in addressing the primary goal of label placement which is to preserve clarity and legibility of names by satisfying well-known cartographic precepts of good lettering on maps on the one hand and on the other hand to maximize the number of labels. This subtask is considered as an optimization problem. Therefore, several compelling strategies to find a feasible near-optimal labelling were proposed in previous research. In particular, these strategies and techniques are: greedy best-first search algorithm (Yoeli, 1972), a discrete gradient-descent method (Hirsh, 1982), exhaustive search algorithms (Ahn and Freeman, 1984; Freeman and Ahn, 1987; Jones, 1989; Cook and Jones, 1990; Doerschler and Freeman, 1992), 0-1 integer programming (Zoraster, 1986), simulated annealing (Christensen et al., 1995; Zoraster, 1997), a depth-first search (Christensen et al., 1995), a genetic algorithm (Verner et al., 1997), an ant colony system (Schreyer and Raidl, 1997), tabu search algorithm (Yamamoto et al., 2002), and POPMUSIC – partial optimization meta-heuristic (Taillard and Voss, 2001; Alvim and Taillard, 2009).

The new algorithm presented in this paper extends previous research by providing techniques for label position generation and evaluation for the task of external labelling of areas. The output of the proposed technical approach can be further utilized as an input for *position selection* subtask to find a good approximation to the global optimum of a general name placement (Edmondson et al., 1996).

6.3 Approach Methodology

In the following subsections we give an exhaustive description of the procedures for label candidate position generation and position quality evaluation, that are compliant with cartographic requirements for annotating polygons externally. An overview of the approach presented in this paper is shown in Figure 6.3.

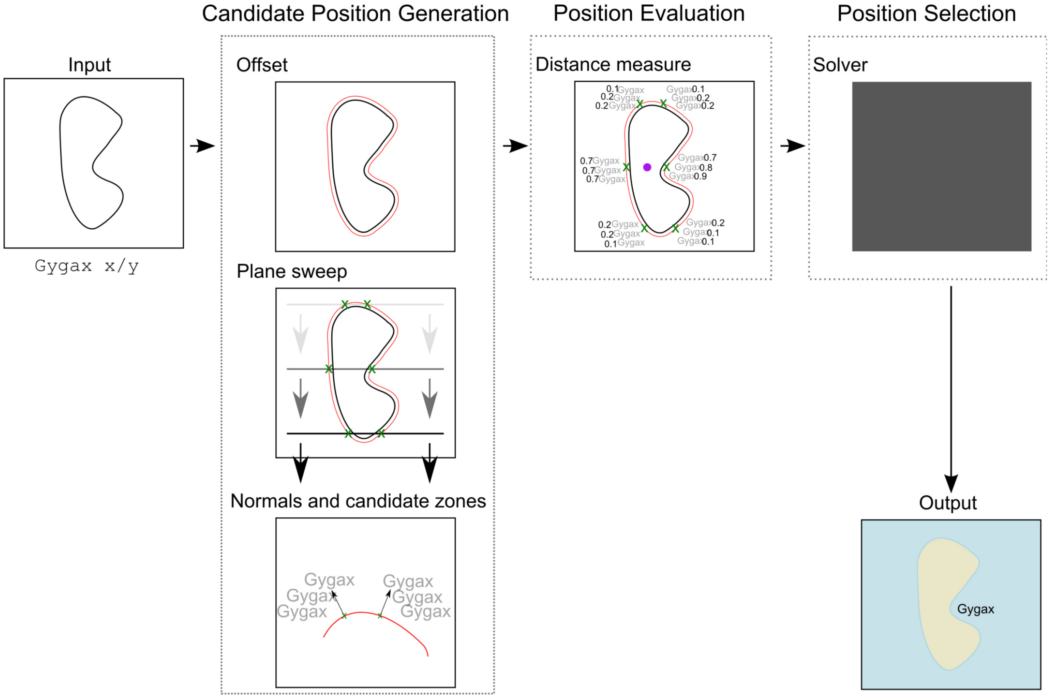


Figure 6.3. Workflow of the presented approach.

6.3.1 Cartographic Guidelines

In order to formalize the criteria for our algorithm we studied well-established and broad cartographic guidelines for external labelling of areas found in the literature (Imhof 1962, 1975; Wood, 2000). The list of rules adapted to the needs of externally labelling areal features is as follows:

R1. Labels should be placed horizontally.

R2. Label should be placed entirely outside at some distance from the area feature.

R3. Name should not cross the boundary of its area feature.

R4. The name should be placed in way that takes into account the shape of the feature by achieving a balance between the feature and its name, emphasising their relationship.

R5. The lettering to the right and slightly above the symbol is prioritized.

In the following subsections we utilize four of the five rules for two subtasks of label placement, namely, for candidate positions generation (R1, R2, and R3) and for measuring their “goodness” (R4). The rule R5 is applicable only in the case when the area of a polygonal feature is small and the feature can be treated and labelled as a point-feature. For a detailed study on the issue of lettering point-features for small and medium scale maps please refer to Wu and Buttenfield (1991).

6.3.2 Candidate-position Generation

Before starting the description of our method we provide the following definitions. The input of our algorithm consists of a non-self-intersecting polygon $Q = (q_1, \dots, q_m)$ in the Euclidean plane $\mathbb{R}^2$. The polygon Q is specified by a sequence of points $q_i = (x_i, y_i)$, where $i = 1, \dots, m$, (see Figure 6.5) and the vertices q_i that are ordered counterclockwise (CCW) around Q's interior. Let L be a label (axis-aligned rectangle) in $\mathbb{R}^2$ that we want to place somewhere outside polygon Q. A desired positioning of L, which satisfies rule R3, can be written as $Q \cap L = \emptyset$.

6.3.2.1 Polygon Offset

We wish to place a label at some distance from the polygon (R2). For that we first construct a bounding polygon that fully contains the original one. For example, this task can be solved by employing one of the properties of the *Minkowski sum* (Lee et al. 1998; Agarwal et al., 2002) of a simple polygon in $\mathbb{R}^2$. This operation is also known as *offsetting* the polygon (Chen and McMains, 2005; Wein, 2007; Bo, 2010). The task of offsetting the polygon (or polyline) is a fundamental geometric problem in variety of applications such as robot motion, computer-aided design and manufacturing, and cartography (see Figure 6.4). It is known that if only one of the polygons is convex then the Minkowski sum of them is bounded by $O(mn)$, where m and n is the number of vertices in the polygons respectively. The computational complexity for finding the Minkowski sum of a polygon with n vertices with a disc is $O(n)$. It is worth noting that for general purpose of our algorithm both exact (Agarwal et al., 2002; Flato, 2000) and approximate algorithms (Wein, 2007) for offsetting a polygon can be utilized.

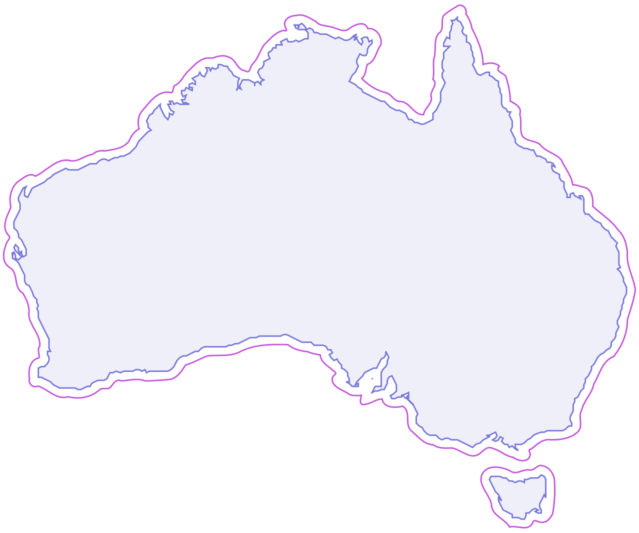


Figure 6.4. A sample polygon offsetting built for Australia continent and Tasmania Island.

6.3.2.2 Plane Sweep Algorithm

Assume that we managed to find the bounding polygon $P = (p_1, \dots, p_n)$ for the original one Q (see Figure 6.5). We now turn our attention to finding candidate positions that lie on the boundary of the polygon P.

We construct a set H of k *horizontal lines* that are equally distributed from the maximum y-value of P to the minimum y-value. The distance d between horizontal lines is an input parameter. Then we want to determine all intersection points among the segments $S = \{s_1, s_2, \dots, s_n\}$ of the polygon P and the k horizontal lines. These intersection points are potential candidate positions for our algorithm.

In computational geometry such a task is well-known and called "*line segment intersection problem*" (de Berg et al., 2008). At first sight, this problem does not seem like a challenging one. The brute-force algorithm is able to find all intersection points of two sets of segments and requires $O(N^2)$ time in the worst case, where $N = n + k$. But it is obvious that the "naive" algorithm is excessive in this case, as most segments have no or only few intersections with segments from another set. One of the first attempts to present a faster algorithm was proposed by Bentley and Ottmann (1979) that extended an idea of Shamos and Hoey (1976). Bentley and Ottmann gave an algorithm that reports all intersections of N segments in $O\big((N + i)\log N\big)$ and needs $O(N + i)$ storage space, where i denotes the number of computed intersections. After many years in spite of presence of slightly faster algorithms (e.g., $O(N \log N + i)$ time and $O(N + i)$ space algorithm of Chazelle and Edelsbrunner (1992) or $O(N \log N + i)$ time and $O(N)$ space algorithm of Chan (1994)) the algorithm of Bentley and Ottmann is still very popular as it is easy to both understand and implement.

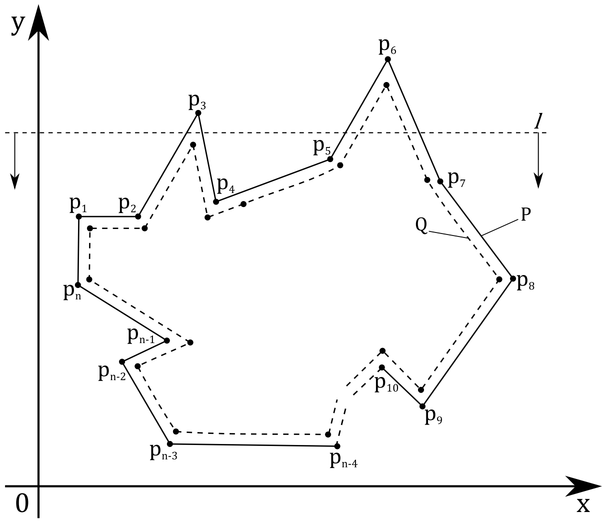


Figure 6.5. The original Q and bounding P polygons; l is a sweep line that moves from top to bottom.

For our task we are going to exploit an idea of Shamos and Hoey (1976) or Bentley and Ottmann (1979) which is the *plane sweep algorithm.* The idea of *sweep line* paradigm is moving of an imaginary horizontal line downwards (or vertical line, the choice is arbitrary) over the plane (see Figure 6.5), starting from the maximum y-value of P and solving the problem (finding intersection points) as it moves. The status of the sweep line is the set of polygon segments which intersect. The implementation of such an algorithm requires two data structures, namely *event queue* EQ and *status (or data) structure* R (Figure 6.6). The so-called event queue contains the events where the sweep line stops. The original version of the algorithm distinguishes three types of events: *upper endpoint*, *lower endpoint*, *intersection.* The status structure contains all segments that intersect the sweep line at each point of the sweep. It also required that the status structure is maintained as a *balanced binary search tree* (Cormen et al., 2009). Hence, the time for performing required operations is $O(\log N)$.

After we gave a brief review of Bentley and Ottmann's algorithm we will now present a modified version of their algorithm to solve the task that was stated in the first paragraph of the section. We will show that the special case we are interested in requires the total running time less then $O\big((n+k+i)\log(n+k)\big)$, where $n+k$ is the number of line segments and i is the number of intersecting pairs.

The main loop of our algorithm sweeps a horizontal line from top to bottom through the endpoints of the segments in S. Each time when an upper (respectively lower) endpoint of a segment s is scanned the segment s is inserted into (removed from) R, where the segments are ordered by their x-coordinates. When a lower endpoint is encountered we also compute intersection points of the horizontal lines H,whose y-values

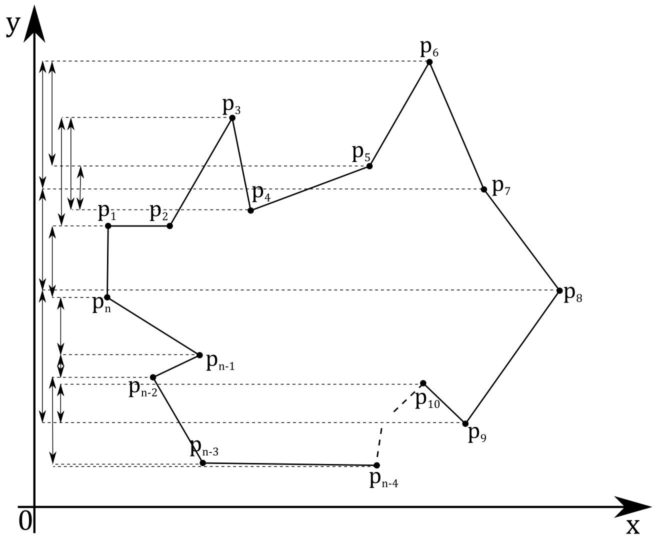


Figure 6.6. Orthogonal projections of the polygon segments onto the y-axis.

Algorithm 1: *FindSegmentLinePairs*

Input: A set S of line segments and a space d between horizontal lines.

Output: The set of pairs, where each pair consists of a segment and a corresponding horizontal line.

```
begin
    Initialize an empty event queue $EQ$. Insert the segment endpoints $p_i$ into $EQ$,
        and store them in descending $y$-order.
    Initialize an empty status structure $R$.
    Let $y_{\mathrm{max}}$ be the maximum y-value of the segment endpoints.
    Let $y_{\mathrm{p}}$ be a previous $y$-value of a horizontal line.
    $y_{\mathrm{p}} \leftarrow y_{\mathrm{max}}$
    foreach $p$ in $EQ$ do
        if $p$ is the upper endpoint of the line segment $s$ then
            Insert $s$ in $R$.
        end if
        else if $p$ is the lower endpoint of the line segment $s$ then
            Let $y_{\mathrm{c}}$ be the $y$-value of $p$.
            if $y_{\mathrm{p}}$ not equals $y_{\mathrm{c}}$ then
                Let $y_{\mathrm{h}}$ be the $y$-value of a current horizontal line.
                $n \leftarrow (y_{\mathrm{p}} - y_{\mathrm{c}})/d$
                $j \leftarrow 1$
                if $y_{\mathrm{p}}$ equals to $y_{\mathrm{max}}$ then
                    $j \leftarrow 0$
                for $i \leftarrow j,\ n$ do
                    $y_{\mathrm{h}} \leftarrow (y_{\mathrm{p}} - \mathrm{d} * \mathrm{i})$
                    foreach segment $t$ in $R$ do
                        if $y_{\mathrm{h}}$ less or equal to $top(t)$ and $y_{\mathrm{h}}$ greater or equal to $bot(t)$ then
                            Set that $t$ was processed.
                            Report the pair $(t,\ y_{\mathrm{h}})$.
                        end if
                    end foreach
                end for
                $y_{\mathrm{p}} \leftarrow y_{\mathrm{h}}$
            end if
            if $s$ was not processed then        //See case d) in Section 6.3.2.3
                Report the pair $(t,\ y_{\mathrm{c}})$.
            end if
            Remove $s$ from $R$.
        end if
    end foreach
end
```

lies between the y-values of the current and previous lower endpoint respectively, with the segments from R. We describe this algorithm in pseudo-code as given in Algorithm 1.

The difference from Bentley and Ottmann's algorithm is that we omitted the intersection event type and we did not consider k horizontal lines in the input as they are "predefined" and can be computed during runtime. This modification has allowed us to construct an algorithm that is able to report all intersecting pairs in $O(n\log n + i)$ time, where i is a number of segment-horizontal line pairs.

It is clear that initialization and processing of $2n$ events requires $O(n\log n)$ time. Reporting all i pairs can be done in time proportional to their number. This assumption is similar to the one that was made in Algorithm 1 in the work of Bentley and Ottmann.

6.3.2.3 Computation of Candidate Positions

We now focus on the computation of label candidate positions. The result of Algorithm 1 is a set of pairs where each pair consists of a horizontal line and a polygon segment. There are three cases when they can intersect each other (see cases **a**)-**c**) in Figure 6.7) and one case (**d**) when they are almost collinear:

- **a.** A horizontal line intersects a segment in its vertex.
- **b.** A horizontal line intersects a segment somewhere in internal part.
- **c.** A segment coincides with a horizontal line.
- **d.** A segment is almost collinear to a horizontal line.

The case **c**) represents a degenerated case when there are infinitely many points that can be referred to the intersection points c_j. In order to limit the number of points we propose to compute additional points that lie on a horizontal line and are evenly spread between two vertices of a segment. Note that the same approach can be used for the case

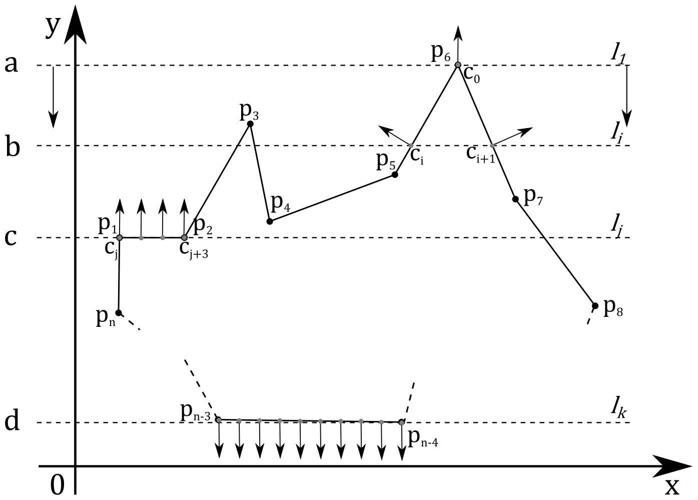


Figure 6.7. Generation of potential label positions.

d). On the one hand this approach allows increasing the number of potential candidate positions; on the other hand it enhances our algorithm in x-direction. The space between evenly generated points can be set to the distance between horizontal lines. The following pseudo-code (Algorithm 2) describes our idea:

Algorithm 2: *FindCandidatePositions*
Input: A set S of ordered line segments and a space d between horizontal lines.
Output: The set of tripl p, a, h where p is the coordinates of a point, and a is an angle of the normal and h is a horizontal line.

```
begin
    foreach $\tau$ in FindSegmentLinePairs($S$, $d$) do
        Let $h$ be a horizontal line in $\tau$.
        Let $s$ be a segment in $\tau$.
        if $h$ and $s$ are collinear or almost collinear then
            Compute the normal $n$ to the $s$.
            Let be $a$ the angle of $n$.
            foreach $v$ in GeneratePointsAlongSegment($s$, $d$)
                Report the position $(v, a, h)$.
            end foreach
        end if
        else
            Let $c$ be a point of intersection between $s$ and $h$.
            Compute the normal $n$ to the s using also the neighbors from $S$.
            Let be $a$ the angle of $n$.
            Report the position $(c, a, h)$.
        end else
    end foreach
end
```

Now we turn to the task of how to use the intersection and additional points c_j as potential positions for labels. For that we utilize a technique that is used for labelling point-features. We consider each point c_j that is an output of Algorithm 2 as a point-feature and detect a set of possible label positions (see Figure 6.8) around the point that lie outside and are therefore unlikely to overlap with the polygon.

To accomplish this task we need to determine a normal to P in each point c_j (see Figure 6.7). As we required that the original polygon is ordered counterclockwise, a normal that is directed outside can be found explicitly. After the normal is computed we can

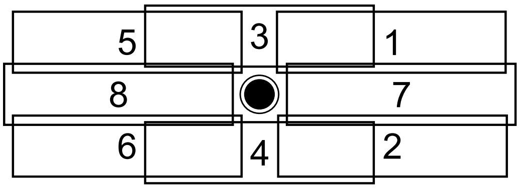


Figure 6.8. Annotation scheme of a point-feature with numbered positions (Imhof 1975).

find label positions (see Figure 6.8) by means of zones that are presented in Figure 6.9. Thus, at each point c_j can be up to 3 label candidate positions.

6.3.2.4 Polygon and Label Intersection

The final step of the candidate position generation is to check whether each potential label, with a position that has been found in the previous section, intersects P or not (R3). Although we constructed potential label positions in such a way that each label does not intersect the polygon segment where its anchor point lies (section 6.3.2.3), the label has a chance to intersect other segments.

Let us assume that we have m label positions. Assuming each label is represented by a rectangle, a "*brute-force*" algorithm for testing whether m rectangles intersect polygon P takes $O(4mn)$. According to Theorem 5 of Shamos and Hoey (1976), to check whether two simple plane polygons intersect can be determined in $O(N \log N)$, where N is the total number of line segments in two polygons. Hence, our task can be accomplished in $O(m(n+4)\log(n+4)) = O(mn \log n)$ time. Furthermore, our case does not require testing whether one of the polygons wholly contains another, which in addition takes $O(N)$ time. This is due to the fact that we generated labels outside the polygon. We also note that if $n+4$ is very close to 10,000 then the "*brute-force*" $[O(4mn)]$ approach becomes faster.

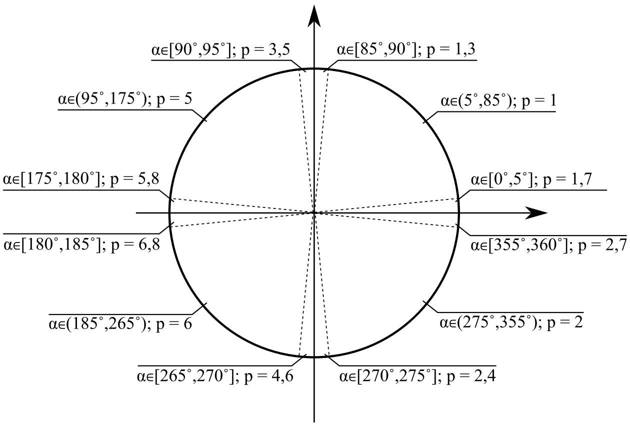


Figure 6.9. Zones of point-feature label positions.

Now we want to describe an algorithm that requires less than $O(mn \log n)$ time for any n. For our algorithm we need two data structures. The first structure is a binary search tree T (Cormen et al., 2009) that stores projections of the segments of the gon P onto y-axis. The second one is a list of polygon segments, denoted by V, which currently can intersect a label candidate. Initially this list is empty. We also need to store a y-coordinate of the previous horizontal line in y_p which is initialized with $-\infty$. The main loop of the algorithm iterates through each point that has been received with the help of Algorithm 2. If the y-coordinate y_c of any candidate position does not equal y_p we clear V and fill it with segments from T that lie within the interval $(y_\text{c} - h, y_\text{c} + h)$, where h is the pre-computed height of the text. And we also assign y_p to y_c. Note that we make a query to T only once for each horizontal line. Further, we generate label candidate positions computed according to the zones given in Figure 6.9. For each label candidate position we extract a bounding rectangle and check if it intersects any segment in V. If it does not, we return a candidate position as the result. We include the present-

Algorithm 3: *FindLabelCandidatesOutsidePolygon*
Input: A set S of line segments, a space between adjacent horizontal lines d and a text height h.
Output: The set of label positions.

```
begin
    Initialize binary search tree $T$ with $y$-projections of the line segments.
    Initialize an empty list of segments $V$.
    Let $y_\text{p}$ be a previous $y$-value of a horizontal line.
    $y_\text{p} \leftarrow -\infty$
    foreach $p$ in FindCandidatePositions ($S$, $d$) do
        Let $y_\text{c}$ be the $y$-value of the horizontal line in $p$.
        if $y_\text{c}$ not equals to $y_\text{p}$ then
            Clear $V$.
            Fill $V$ with segments from $T$ that lie within the interval $(y_\text{c} - h, y_\text{c} + h)$.
            $y_\text{p} \leftarrow y_\text{c}$
        end if
        foreach $l$ in FindLabelPositionsByZones($p$)
            Let $b$ be the bounding rectangle of the label $l$.
            if $b$ not intersects $V$ then
                Report the label placement ($l$).
            end if
        end foreach
    end foreach
end
```

ed algorithm in pseudo-code as Algorithm 3.

We omit the description of the function *FindLabelPositionsByZones* due to its simplicity. We will now examine the complexity of the proposed Algorithm 3. The initialization of the structure T requires $O(n \log n)$. Further, our algorithm consists of k operations of retrieving segments from T which takes $O(k \log n)$. Additionally, the process of removing and inserting segments from/into the list V takes $O(2kn) = O(kn)$ time. As the function *FindLabelPositionsByZones* (see Algorithm 3, line 13) is able to generate m candidate positions then we need to perform up to m checks whether the bounding rectangle of a label intersect P in the worst case n segments. This can be done in $O(mn)$ time. Hence, the total performance time of the algorithm is $O\big((n+k)\log n + (k+m)n\big)$. However in practice, the true complexity of the algorithm is a function of the spatial distribution of the vertices in a polygon. For a given compilation scale/level of detail in the realm of geographic data, this distribution can usually be interpreted as being constant.

6.3.3 Position Quality Evaluation

The quality evaluation component of our algorithm evaluates the goodness of a label candidate position in respect to the degree of a spatial relationship between a label and the feature it tags (R4). The higher degree of their relationship the faster the information is searched and interpreted by the map reader (Lloyd, 1997). The distance between two spatial objects is normally measured by some proximity measure. A wide range of different proximity measures are extant in the literature (for instance, Laube et al., 2008; Zighed et al., 2012). Usually, proximity is expressed as a function to compute a single numeric score in the range [0, 1]. For our needs we use a rather simple proximity measure that is based on measuring the Euclidian distance between the centroid points of the polygon and a label. In order to convert a distance to a score value that falls into in the range [0, 1] we define the quality function of a label l as:

$$Q(P,l) = \frac{U_{\mathrm{e}}(C(P), C(l))}{\max\limits_{l \in L} U_{\mathrm{e}}(C(P), C(l))},$$

where $C(P)$ is a function for computing the centroid point for a polygon P or bounds of a label l, L is a set of all label candidate positions of the polygon P and U_{e} is the Euclidian distance:

$$U_{\mathrm{e}}(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}.$$

It is worth noting that in our measure we choose a linear score function, however when so required, one can also use supposedly more realistic and suitable quantification functions, for example, a non-linear or a smooth Gaussian-like function.

As mentioned above, our approach actually reduces the problem from the more involved and ambiguous polygon to polygon proximity measurement (Laube et al., 2008) to a point to point proximity question. This was done mostly to keep the overall complexity low. In order to accomplish this reduction, we need a point that visually best represents the whole respective polygon. While this seems easy enough for the rectangular label, for concave polygons the question is known to be strongly dependant on the use case (Carstensen, 1987). Where centre of gravity has been shown to be a bad approximation for diagram symbol placement as it can often lie outside the polygon, for external labelling this naturally is not a problem. In fact, according to cartographic practice and literature (Wood, 2000), C- or U-shaped polygons can best be served by placing their label into the 'bay' (conceptual or real). Using the distance between the centroids of both label and area feature as the proximity measure naturally guides the label into such 'bays' with the proximity value of zero being preferred and reachable for strongly concave polygons.

6.4 Implementation and Experimental Results

This section presents the results of the computational tests of our practical approach and comparison with other labelling engines.

6.4.1 Implementation

We have implemented a version of the algorithm presented in this paper within a framework for publishing spatial data to the web. This framework is written in C# and called *MapSurfer.NET.* We ran our experiments on a machine with an Intel® Core™ i5-2500 CPU @ 3.30 GHz running Windows 7 Professional x64 with 8GB installed memory. The runtime execution environment of our test application was .NET Framework 4.5 (x64).

In our implementation of the algorithm we utilized the *Clipper* library for polygon offsetting. The distance between horizontal lines d is normally defined by the user which can control the size of the *search space* in the *point selection* task, which has a dramatic effect on computational complexity of a general labelling algorithm. Further, in order to increase the number of candidate positions for small polygons (number of horizontal lines $k < 5$) we compute the distance d as follows: $d(P) = (y_{\max}(P) - y_{\min}(P))/4$. The offset for constructing bounding polygon is equal to 6 map units (pixels in our tests).

6.4.2 Dataset Information

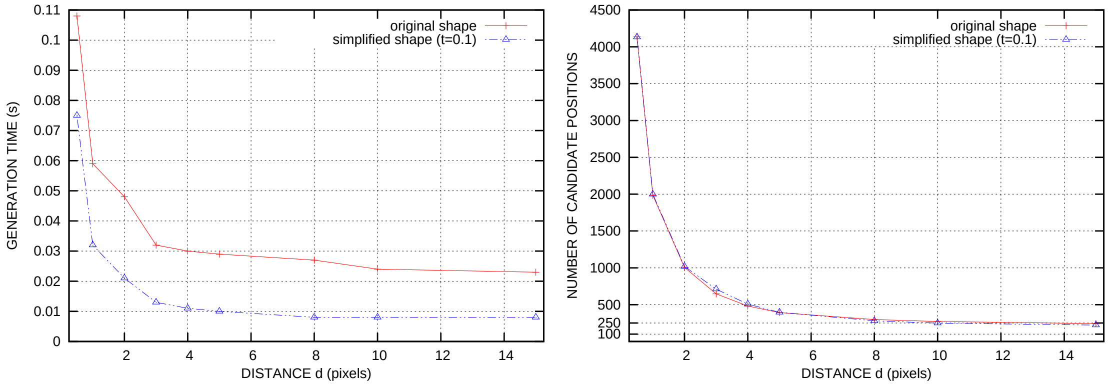
The test dataset was extracted from the dataset of *OpenStreetMap* project that is one of the most promising crowd sourced projects for collecting geospatial data. The area of interests represents the northwestern part of the Netherlands and Germany, namely the Frisian Islands which are a group of islands in the Wadden Sea, part of the North Sea.

6.4.3 Experimental Results

We carried out the experiments on the North Frisian Islands. The tests were run on two sets of polygons (19 features) with 3650 and 1269 vertices in total respectively. The first one is the original polygons that were taken from the data source. The second set was derived from the first one by applying the polygonal simplification algorithm of Ramer (1972), Douglas and Peucker (1973) with the tolerance distance equals to 0.1 of map units. Figure 6.10a shows the running time of the algorithm for generation and evaluation of candidate positions with different values of d on x-axis. Figure 6.10b depicts relation between the distance d and the number of generated label positions. It is clear from the figures that the running times as well as the number of candidate positions grow linearly when the space between horizontal lines is decreased. Our algorithm shows a rather good performance and in average is able to produce one label placement in $6.25 \cdot 10^{-5}$ seconds for the set of original polygons and in $2.49 \cdot 10^{-5}$ seconds for their simplified version. Hence the simplification of polygons gains 2.5 times improvement in speed (see Figure 6.10a) with almost the same number of candidate positions in the output (see Figure 6.10b).

Now we turn to the task of evaluating the label positions. We computed numeric values that indicate labelling quality using the metric given in Section 6.3.3. Then we mapped the scalar values to colours (see Figure 6.11) and visualized them as filled circles on top of the map without labels of islands. It can be seen from Figure 6.11 that the results visually give a very descriptive picture of a good and bad positions for labelling of a certain map feature. The regions that are coloured to red or yellow colours indicate most preferable positions.

A resulting map is shown in Figure 6.12 which contains all feature types (points, lines and areas) involving in map annotation. The map lettering was made by using a *simulat-*



(a) (b)

Figure 6.10. Results of candidate position generation with different distance between horizontal lines: (a) Running time; (b) Number of generated positions.

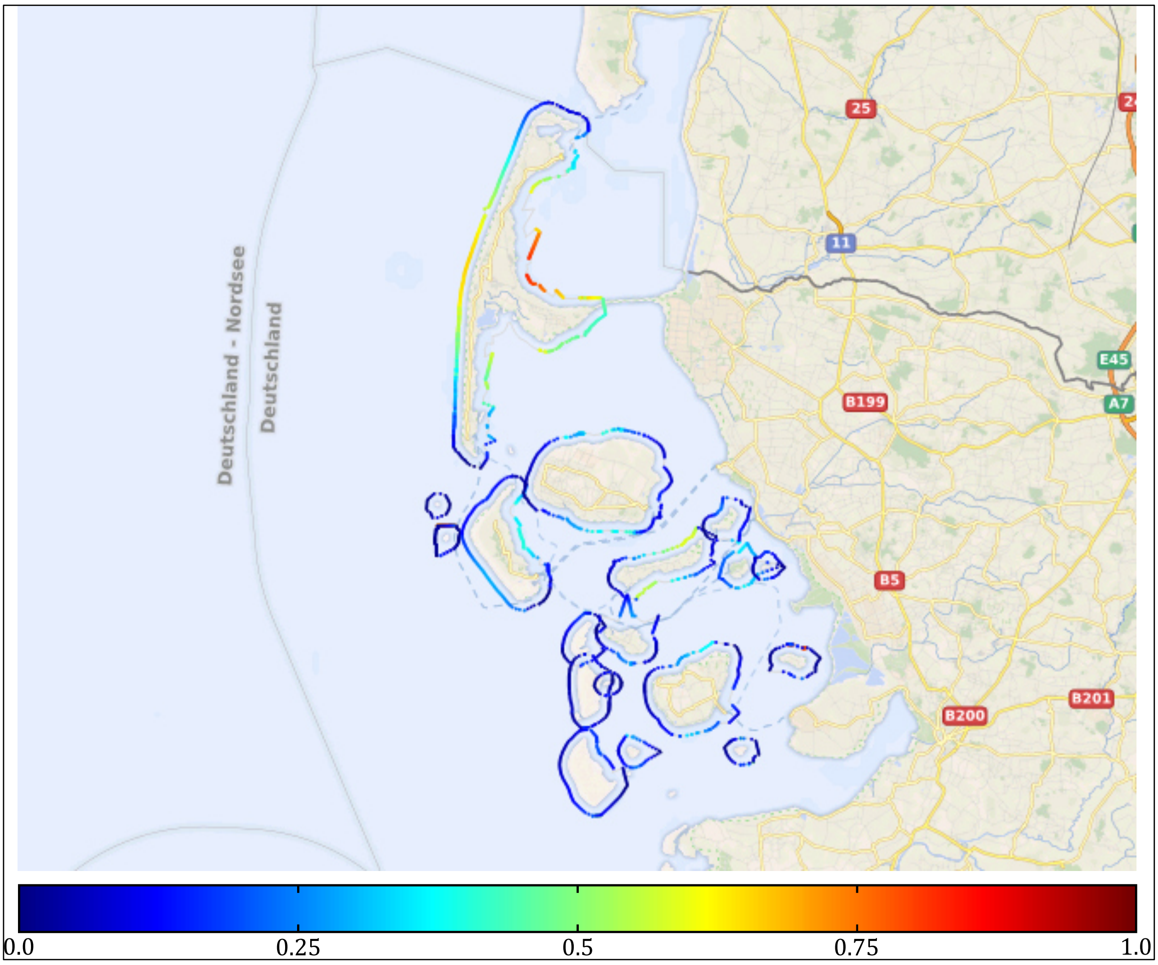


Figure 6.11. Displaying of candidate positions by mapping their scalar quality values to colours ($d = 1, r = 6$). Data source: OpenStreetMap project (2013).

ed annealing algorithm (Christensen et al., 1995; Zoraster, 1997) to find a near-optimal solution of the label placement problem which is described through a multi-criteria model for cartographic label placement. For generating and evaluating label positions for the islands we used the algorithm presented in this paper. In Figures 6.13 and 6.14 the reader can compare two maps of the same region. The first map (Figure 6.13) was annotated using our algorithm, and the second one (Figure 6.14) was labelled manually by a cartographer. The two figures highlight that automated name placement with our algorithm and parameterization is very close to the manual lettering done by a skilled map designer. In summary, Figures 6.12 and 6.13 demonstrate the ability of our algorithm to produce a plausible labelling of areas outside their boundaries on small and medium scales.

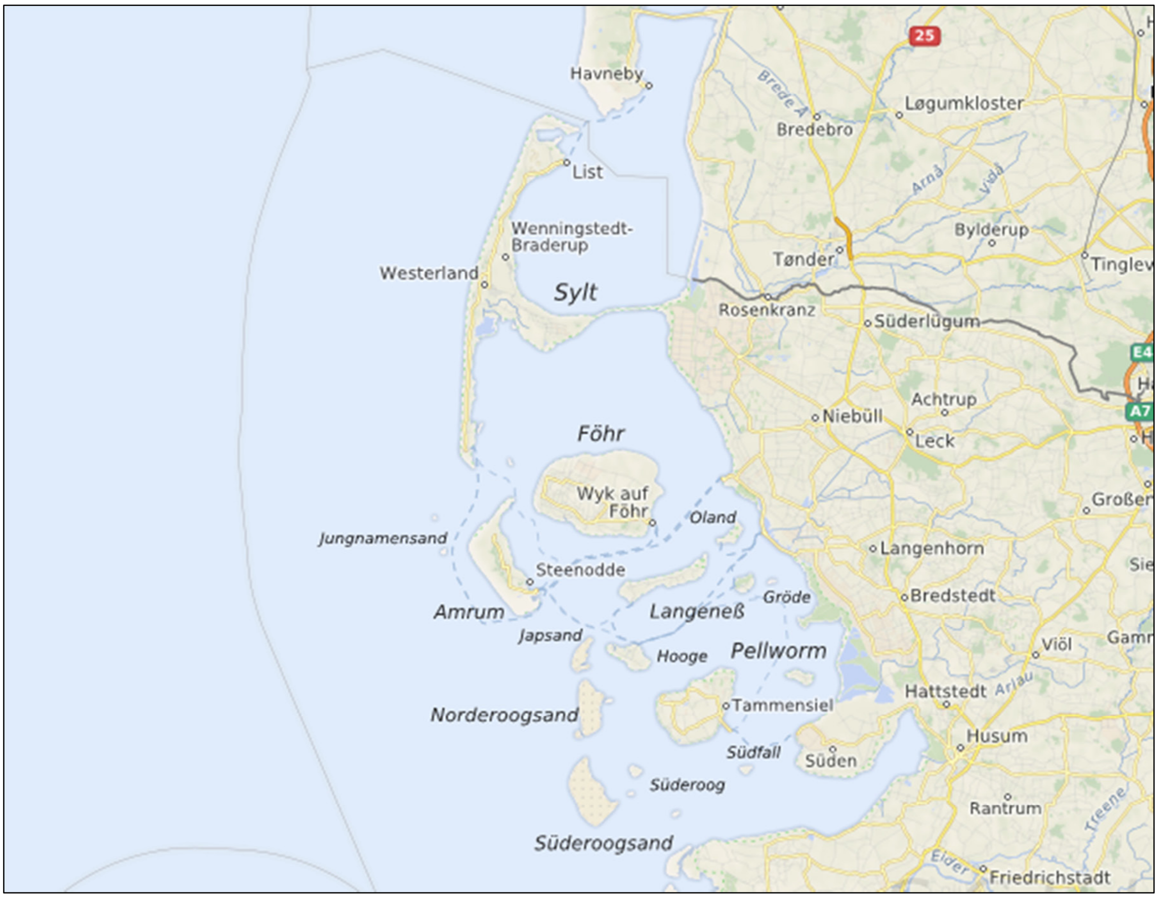


Figure 6.12. A resulting map with a labelled group of the North Frisian Islands and other feature types. Projection: spherical Mercator (EPSG:3857). Data source: OpenStreetMap project (2013).

6.4.4 Comparison with Existing Label Placement Engines

In order to demonstrate the efficiency and the distinctiveness of our method, we made a visual comparison of the label placement produced by the presented algorithm with the lettering performed by Maplex and Label-EZ. Note that a runtime comparison is not possible as the proprietary packages internally embed and couple the candidate position and evaluation functions with the position selection procedure. We compared three different labelling algorithms on the West and East Frisian Islands, more precisely on the 20 islands that are located between Terschelling and Wangerooge islands inclusively. The same input data (e.g., shape of islands, font style and font size) were used for all tested solutions. Maplex has been configured according to the preferences depicted in Figure 15. Note that the positional prioritization given in Figure 6.15 is always applied globally, i.e. does not take into account the shape and orientation of the labelled polygons at all. As

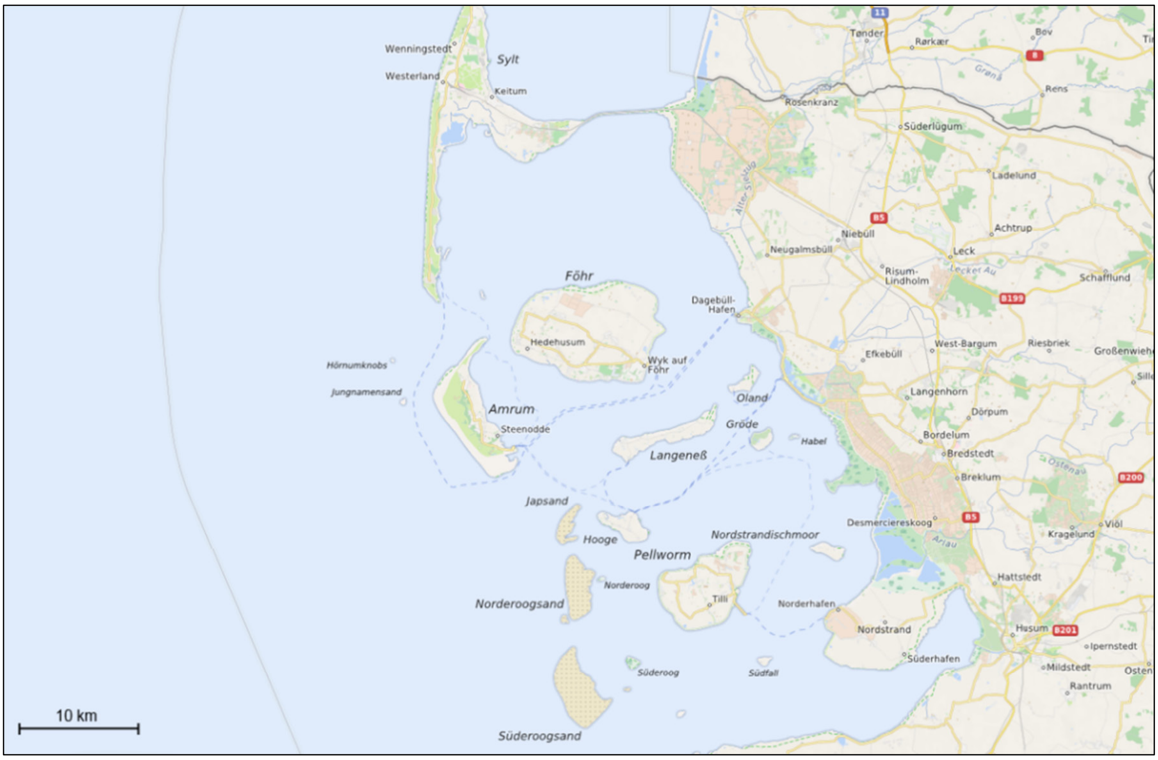


Figure 6.13. A resulting map with a labelled group of the North Frisian Islands and other feature types. Projection: spherical Mercator (EPSG:3857). Data source: OpenStreetMap project (2013).

such, any optimization of the preferences will only ever be locally appropriate. The preferences used in the test were chosen to highlight this shortcoming. Label-EZ does not have any additional options for generating candidate positions.

Figure 6.16 shows the resulting label placement produced by each competing algorithm. Maplex (Figure 6.16a) and the proposed algorithm (Figure 6.16c) placed labels for all islands, whereas Label-EZ (Figure 6.16b) was not able to label Engelsmanplaat Island for unknown reasons. It can be seen from Figure 6.16a that Maplex first tries to locate labels at the left upper zone of an island (see Figure 6.15), which in several cases vastly contradicts rule R4. For example, visually plausible and cartographically preferred positions for labelling Borkum and Langeoog would be bights of the islands. Label-EZ is also not able to produce lettering of with a satisfactory degree of graphic association between the names and the corresponding features. The map produced by this engine does not have such pronounced uniform oriented labelling as Maplex. However, the location of lettering for some islands (e.g., Ameland, Schiermonnikoog and Borkum) is far from their ideal positions. Figure 6.16c demonstrates that our approach outperforms the other test-

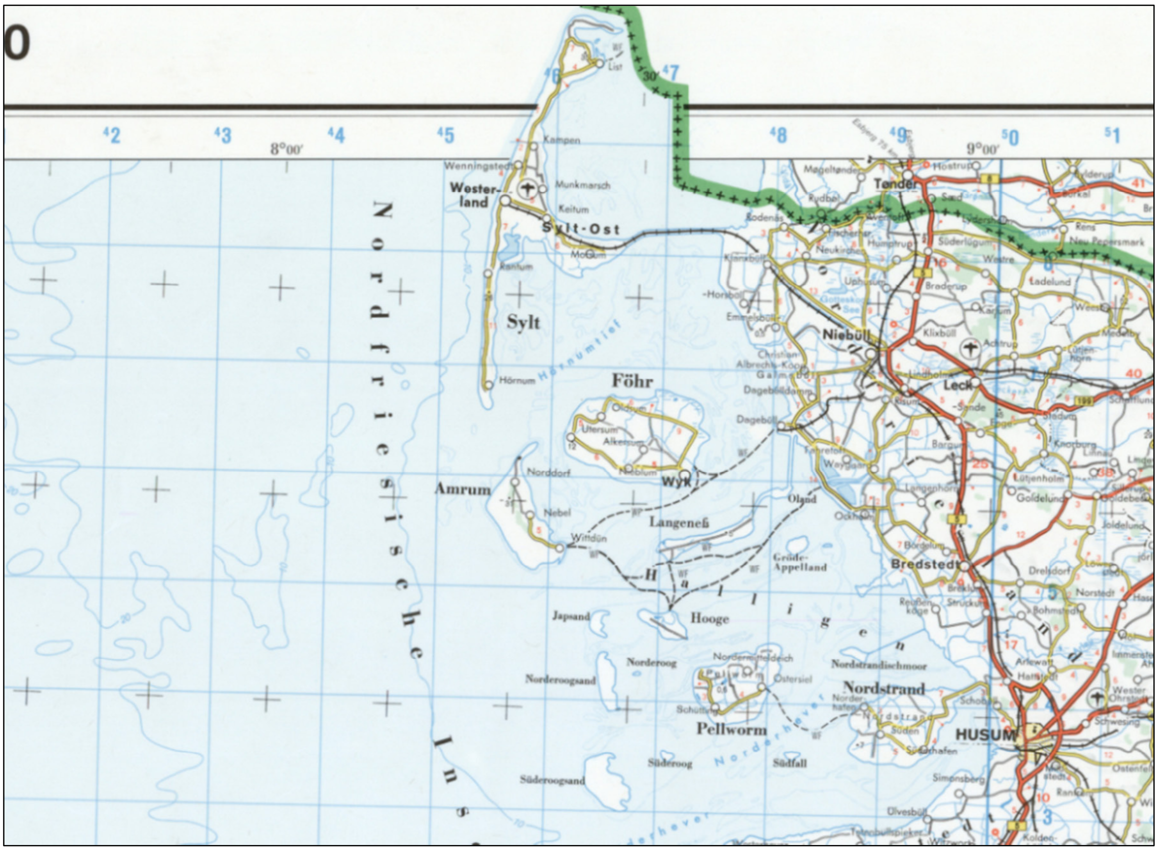
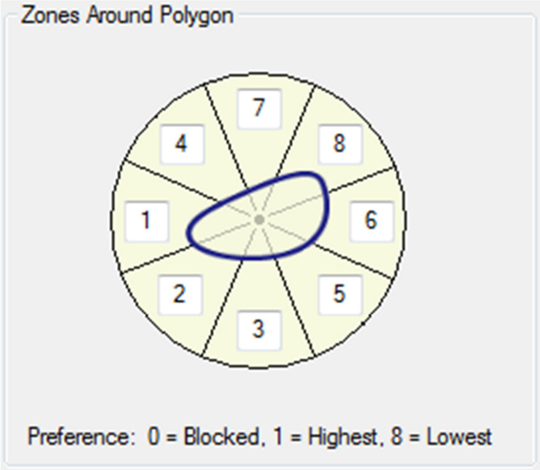
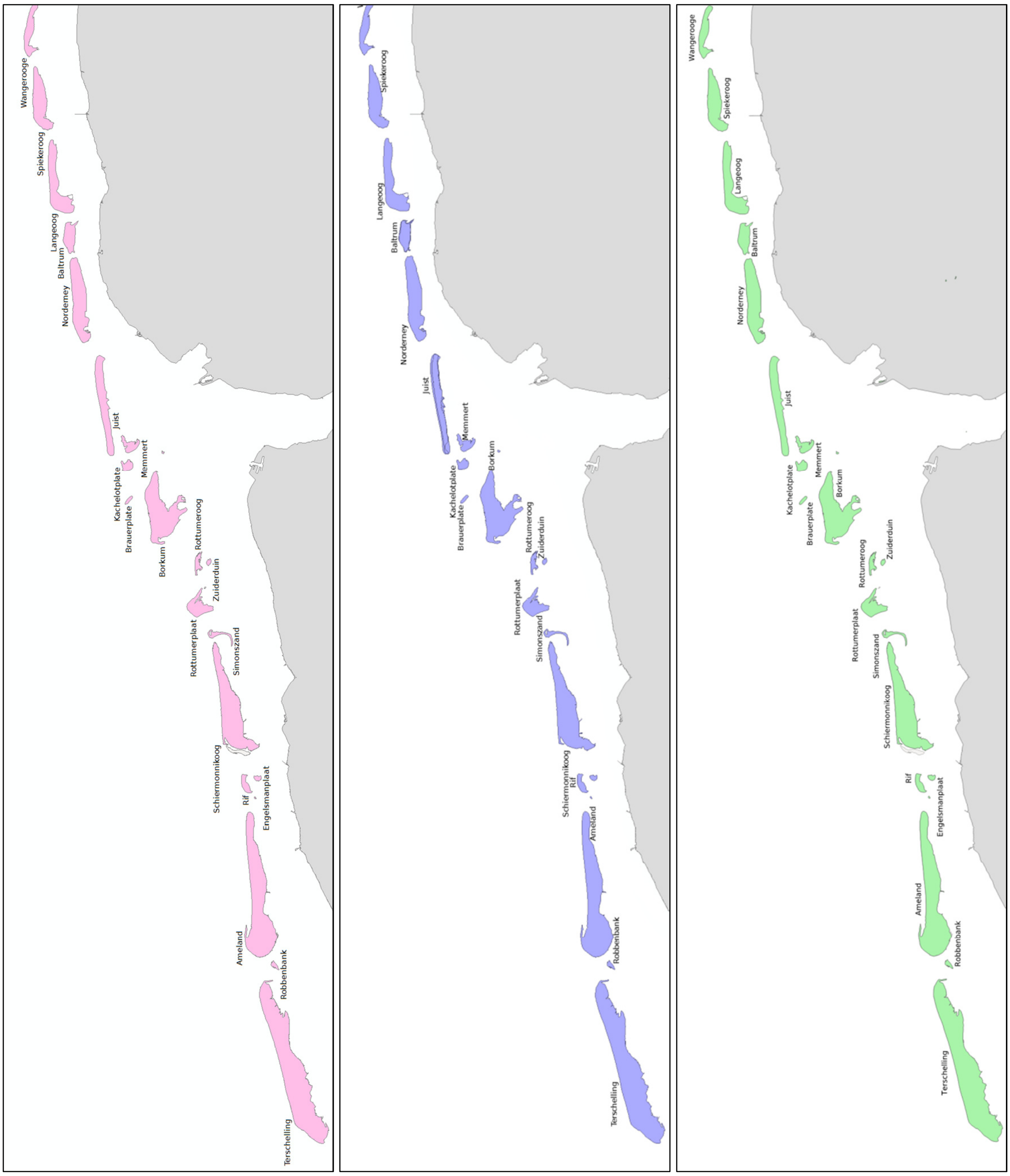


Figure 6.14. The North Frisian Islands. Projection: Equidistant conic projection (standard parallels 49 ° and 56 °), scale 1:500000. Source: Deutscher Militärgeographischer Dienst (1990), © BGIC – Licence B-14A003.


Zones Around Polygon

Preference: 0 = Blocked, 1 = Highest, 8 = Lowest

Figure 6.15. The preferences used by Maplex that are taken into account when placing a label at the best position around a polygon.



(a) (b) (c)

Figure 6.16. Comparison of type placement produced by different labelling algorithms. Projection: spherical Mercator (EPSG:3857). Data source: OpenStreetMap project (2013). (a) Maplex; (b) Label-EZ; (c) Proposed method.

ed methods in quality. In contrast to the labelling produced by Maplex and Label-EZ, the labels for the islands Borkum, Spiekeroog and Langeoog are placed exactly in the bights of the islands. Furthermore, the lettering of horizontally stretched islands such as Terschelling, Ameland, Schiermonnikoog, Juist, Norderney, Baltrum, Langeoog, Spiekeroog and Wangerooge is emphasizing the attribution to their respective polygons.

6.5 Conclusions

This paper presents a novel and highly optimized algorithm for generating candidate positions and an efficient measure for evaluating label positions for areas outside the boundary. Our algorithm copes with a set of cartographic precepts for labelling areas in such a way (Section 6.3.1). In order to fulfill the cartographic requirements we utilized algorithms from the field of computational geometry, namely computing polygon *offsetting* (Section 3.2.1), a *plane sweep* algorithm (Section 3.2.2) and a *balanced binary search tree* (Section 3.2.4) to find label candidate positions. As a result the algorithm produces convincingly good results in terms of performance. For evaluating label positions we proposed a simple metric that consists of measuring the Euclidian distance between the centroid points of the polygon and its label.

The experimental results (Section 6.4.3) show that on the one hand our algorithm has a rather good performance and can be used even in interactive and dynamic visualizations (Been et al., 2006; Zhang and Harrie, 2006; Mote, 2007), on the other hand our straightforward quality measure evaluates candidate positions well enough to produce visually appealing labelling. In most cases the labels and corresponding polygons have an unambiguous relationship. Thus, the labelled objects can be easily identified, which is the primary goal of label placement. Moreover, our comparative study (Section 6.4.4) confirms that the proposed technique outperforms both Maplex and Label-EZ in the quality of name placement for a certain type of features. It is evident from the study that our algorithm incorporates more cartographic guidelines than other existing solutions.

We have implemented the devised algorithm as a part of a map labelling toolkit. This toolkit is used for the rendering of a web map that is based on the dataset provided by *OpenStreetMap* project. All islands on the map on small scales (lower zoom levels) are labeled using the algorithm given in this paper. The map is available online on the *OpenMapSurfer* web page as “OSM Roads” layer.

Acknowledgements

The authors would like to warmly thank colleagues at our institute for their early reviews and valuable feedback. Finally, Maxim Rylov gratefully acknowledges the DAAD

(Deutscher Akademischer Austauschdienst, German Academic Exchange Service) for awarding the DAAD-Scholarship to pursue his doctoral degree.

References

Agarwal, P., E. Flato, and D. Halperin. 2002. "Polygon decomposition for efficient construction of Minkowski sums." *Computational Geometry: Theory and Applications* 21: (1-2): 39-61. doi: 10.1016/S0925-7721(01)00041-4

Ahn, J., and H. Freeman 1984. "A Program for Automatic Name Placement." *Cartographica* 21(2-3): 101-109. doi: 10.3138/0646-Q262-6636-3681

Alvim, A. C. F. and É. Taillard, 2009. "POPMUSIC for the point feature label placement problem." *European Journal of Operatioal Research* 192(2): 396-413. doi: 10.1016/j.ejor.2007.10.002

Army Topographic Support Establishment. 1997. Series ONC, Sheet M-12, Indonesia, 1:1000000, Edition 3, Reprinted by Defence Imagery and Geospatial Organisation in 2002. Scan provided by University of Texas, Perry-Castañeda Library Map Collection. http://www.lib.utexas.edu/maps/onc/txu-pclmaps-oclc-8322829_m_12.jpg (accessed July 4, 2013)

Barrault, M. 2001. "A Methodology for Placement and Evaluation of Area Map Labels." *Computers, Environment and Urban Systems* 25(1): 33-52. doi: 10.1016/S0198-9715(00)00039-9

Battista, G. D., P. Eades, R. Tamassia, and I. G. Tollis. 1994. "Algorithms for Drawing Graphs: An Annotated Bibliography." *Computational Geometry: Theory and Applications* 4(5): 235-282. doi: 10.1016/0925-7721(94)00014-X

Been, K., E. Daiches, and C. Yap. 2006. "Dynamic Map Labeling." *IEEE Transactions on Visualization and Computer Graphics* 12(5): 773-780. doi: 10.1109/TVCG.2006.136

Bentley, J. L. and T. A. Ottmann. 1979. "Algorithms for Reporting and Counting Geometric Intersections." *IEEE Transactions on Computers* C–28(9): 643-647. doi: 10.1109/TC.1979.1675432

Bo, Q. 2010. "Recursive polygon offset computing for rapid prototyping applications based on Voronoi diagrams." *The International Journal of Advanced Manufacturing Technology* 49(9-12): 1019-1028. doi: 10.1007/s00170-009-2454-4

Carstensen, L. W. 1987. “A Comparison of Simple Mathematical Approaches to the Placement of Spot Symbols.” *Cartographica* 24(3): 46-63. doi: 10.3138/GH04-24H7-63T8-167V

Castner, H. W. and R. J. Eastman. 1985. “Eye-Movement Parameters and Perceived Map Complexity—II.” *American Cartographer* 12(1): 29-40. doi: 10.1559/152304085783914712

Chan, T. 1994. “A Simple Trapezoid Sweep Algorithm for Reporting Red/Blue Segment Intersections.” In *Proc. The 6th Canadian Conference in Computational Geometry* : 263-268.

Chazelle, B. and H. Edelsbrunner. 1992. “An optimal algorithm for intersecting line segments in the plane.” In *Proc. The 29-th Ann. IEEE Sympos. Foundations of Computer Science* 39(1): 1-54. doi: 10.1145/147508.147511

Chen, X. and S. McMains. 2005. “Polygon Offsetting by Computing Winding Numbers.” In *ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2005),* September 24-28, Long Beach, California, USA.

Chirié, F. 2000. “Automated Name Placement With High Cartographic Quality: City Street Maps.” *Cartography and Geographic Information Science* 27(2): 101-110. doi: 10.1559/152304000783547902

Christensen, J., J. Marks, and S.M. Shieber. 1995. “An Empirical Study of Algorithms for Point-Feature Label Placement.” *ACM Transactions on Graphics* 14(3): 203-232. doi: 10.1145/212332.212334

Clipper – an open source freeware polygon clipping library (maintained by Angus Johnson). Accessed June 12, 2013. Available at http://www.angusj.com/delphi/clipper.php.

Cook, A.C., and C.B. Jones. 1990. “A Prolog Rule-Based System for Cartographic Name Placement.” *Computer Graphics Forum* 9(2): 109-126. doi: 10.1111/j.1467-8659.1990.tb00384.x

Cormen, T. H., C. E. Leiserson, R. L. Rivest, and C. Stein. 2009. “Binary search trees.” In *Introduction to Algorithms*, 3rd ed., 253-272. Cambridge: MIT Press.

de Berg, M., O. Cheong, M. van Kreveld, and M. Overmars. 2008. “Line segment intersection.” In *Computational geometry: Algorithms and Applications.* 3rd ed. 19-41. Berlin-Heidelberg: Springer.

Deutscher Militärgeographischer Dienst. 1990. Serie M444, Sheet 7, Verkehrskarte Deutschland und Nachbargebiete, 1:500000, © BGIC – Licence B-14A003.

Doerschler, J.S., and H. Freeman. 1992. "A Rule-Based System for Dense-Map Name Placement." *Communications of the Association of Computing Machinery* 35(1): 68-79. doi: 10.1145/129617.129620

Dörschlag, D., I. Petzold, and L. Plümer. 2003. "Placing objects automatically in areas of maps." In *Proc. The 23rd Int. Cartographic Conf. (ICC'03)*: 269-275.

Douglas, D., and T. Peucker. 1973. "Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature." *Cartographica* 10(2): 112-122. doi: 10.3138/FM57-6770-U75U-7727

Edmondson, S., J. Christensen, J. Marks, and S. Shieber. 1996. "A General Cartographic Labeling Algorithm." *Cartographica* 33(4): 13-23. doi: 10.3138/U3N2-6363-130N-H870

ESRI. (2009). 'Maplex for ArcGIS. An ESRI White Paper'. Available at http://www.esri.com/library/whitepapers/pdfs/maplex-for-arcgis.pdf.

Flato, E. 2000. "Robust and Efficient Construction of Planar Minkowski Sums", Master's thesis, School of Computer Science, Tel-Aviv University. Available at http://www.cs.tau.ac.il/CGAL/Theses/flato/thesis/.

Formann, M., and F. Wagner. 1991. "A Packing Problem with Applications to Lettering of Maps." In *Proceedings of the 7th Annual Symposium on Computational Geometry*, 281-288.

Freeman, H., and J. Ahn. 1987. "On the Problem of Placing Names in a Geographic Map." *International Journal of Pattern Recognition and Artificial Intelligence* 1(1): 121-140. doi: 10.1142/S0218001487000096

Freeman, H. 1991. "Computer name placement." In *Geographical Information Systems: Principles and Applications.* ed. D.J. Maguire, M.F. Goodchild, and D.W. Rhind. 445-456, London: Longman.

Goodchild, M.F. 2007. "Citizens as Sensors: The World of Volunteered Geography." *GeoJournal* 69: 211-221. doi: 10.1007/s10708-007-9111-y

Haklay, M., and P. Weber. 2008. "OpenStreetMap – User Generated Street Map", IEEE Pervasive Computing 7(4): 12-18. doi: 10.1109/MPRV.2008.80

Hirsch, S.A. 1982. "An Algorithm for Automatic Name Placement around Point Data." *American Cartographer* 9(1): 5-17. doi: 10.1559/152304082783948367

Hong, F., Z. Zuxun, and D. Daosheng. 2005. "Quality evaluation model for map labeling." *Geo-spatial Information Science* 8(1): 72-78. doi: 10.1007/BF02826996

Imhof, E. 1962. “Die Anordnung der Namen in der Karte.” In *International Yearbook of Cartography*, 93-129, Bonn-Bad Godesberg: Kirschbaum.

Imhof, E. 1975. “Positioning names on maps.” *American Cartographer* 2(2): 128-144. doi: 10.1559/152304075784313304

Kakoulis, K. G. and I. G. Tollis., 2003. “A Unified Approach to Automatic Label Placement.” *International Journal of Computational Geometry and Applications* 13(1): 23-59. doi: 10.1142/S0218195903001062

Kato, T., and H. Imai. 1988. “The NP-completeness of the Character Placement Problem of 2 or 3 Degrees of Freedom.” Record of Joint Conference of Electrical and Electronic Engineers in Kyushu, 11-18.

Klau, G.W. and P. Mutzel. 2003. “Optimal labeling of point features in rectangular labeling models.” *Mathematical Programming* 94(2-3):435-458. doi: 10.1007/s10107-002-0327-9

Laube, P., M. de Berg, and M. van Kreveld. 2008. “Spatial Support and Spatial Confidence for Spatial Association Rules.” In *Headway in Spatial Data Handling Lecture Notes in Geoinformation and Cartography.* ed. A. Ruas, and C. Gold. 575-593. Berlin Heidelberg: Springer. doi: 10.1007/978-3-540-68566-1_33

Lee, I-K., M-S. Kim, and G. Elber. 1998. “Polynomial/Rational Approximation of Minkowski Sum Boundary Curves.” *Graphical Models and Image Processing* 60(2): 136-165. doi: 10.1006/gmip.1998.0464

Lloyd, R. 1997. “Visual Search Processes Used in Map Reading.” *Cartographica* 34(1): 11-32. doi: 10.3138/F342-1217-827R-0777

MapText. 2014. Label-EZ: Automated labeling software. Available at http://maptext.com/labelez.

Marks, J., and S. Shieber. 1991. “The computational complexity of cartographic label placement.” Technical Report TR-05-91, Harvard University, March.

Monmonier, M. S. 1982. Computer-Assisted Cartography: Principles and Prospects. New Jersey: Prentice.

Mote, K. 2007. “Fast Point-Feature Label Placement for Dynamic Visualizations.” *Information Visualization* 6(4): 249-260. doi: 10.1057/PALGRAVE.IVS.9500163

OpenMapSurfer, 2013. (a web map service provided by Heidelberg University; accessed August 2, 2013). Available at http://openmapsurfer.uni-hd.de.

Phillips, R.J., L. Noyes, and R.J. Audley. 1977. “The Legibility of Type on Maps.” *Ergonomics* 20(6): 671-682. doi: 10.1080/00140137708931679

Pinto, I. and H. Freeman. 1996. "The Feedback Approach to Cartographic Areal Text Placement." In *Advances in Structural and Syntactical Pattern Recognition Lecture Notes in Computer Science* 1121: 341-350. New York: Springer. doi: 10.1007/3-540-61577-6_35

Ramer, U. 1972. "An Iterative Procedure for the Polygonal Approximation of Plane Curves." *Computer Graphics and Image Processing* 1(3): 244-256. doi: 10.1016/S0146-664X(72)80017-0

Ramm, F., J. Topf, and S. Chilton. 2010. *OpenStreetMap: Using and Enhancing the Free Map of the World*. Cambridge: UIT Cambridge.

Robinson, A.H., J.L. Morrison, P.C. Muehrcke, A.J. Kimerling, and S.C. Guptil 1995. *Elements of Cartography*, 6th ed. New York: Wiley.

Rylov, M.A., and Reimer, A.W. 2014a. "A Comprehensive Multi-Criteria Model for High Cartographic Quality Point-Feature Label Placement." *Cartographica* 49(1): 52-68. doi: 10.3138/carto.49.1.2137.

Schreyer, M., and G.R. Raidl. 2002. "Letting Ants Labeling Point Features." In Proc. IEEE Congress on Evolutionary Computation (CEC '02) 2: 1564-1569.

Shamos, M. and D. Hoey. 1976. "Geometric Intersection Problems." In *Proc. of the 17-th Ann. Conf. Foundations of Computer Science*: 208-215. doi: 10.1109/SFCS.1976.16

Strijk, T.W., and M.J. van Kreveld. 2002. "Practical Extensions of Point Labeling in the Slider Model", *GeoInformatica* 6(2): 181-197. doi: 10.1023/A:1015202410664

Taillard, É. D., and S. Voss. 2001. "POPMUSIC: Partial optimization metaheuristic under special intensification conditions.", In *Essays and Surveys in Metaheuristics*. ed. C. Ribeiro, P. Hansen. 613-629. Boston: Springer. doi: 10.1007/978-1-4615-1507-4_27

van Dijk, S., M.J. van Kreveld, T.W. Strijk, and A. Wolff. 2002. "Towards an Evaluation of Quality for Names Placement Methods." *International Journal of Geographical Information Science* 16(7): 641-661. doi: 10.1080/13658810210138742

van Roessel, W. 1989. "An Algorithm for Locating Candidate Labeling Boxes Within a Polygon." *American Cartographer* 16(3): 201-209. doi: 10.1559/152304089783814034

Verner, O.V., R.L. Wainwright, and D. Schoenefeld. 1997. "Placing Text Labels on Maps and Diagrams using Genetic Algorithms with Masking." *INFORMS Journal on Computing* 9(3): 266-275. doi: 10.1287/ijoc.9.3.266

Wein, R. 2007. "Exact and approximate construction of offset polygons." *Computer-Aided Design* 39(6): 518-527. doi: 10.1016/j.cad.2007.01.010

Wolff, A., L. Knipping, M.J. van Kreveld, T.W. Strijk, and P.K. Agarwal. 2001. “A Simple and Efficient Algorithm for High-Quality Line Labeling.” *Technical Report UU-CS-2001-44*, Department of Computer Science, Utrecht University.

Wolff, A., and T. Strijk, 2009. “A map labeling bibliography.” Available at http://i11www.iti.uni-karlsruhe.de/~awolff/map-labeling/bibliography/.

Wood, C.H. 2000. “Descriptive and Illustrated Guide for Type Placement on Small Scale Maps.” *Cartographic Journal* 37(1): 5-18. doi: 10.1179/000870400787320851

Wu, C.V. and B.P. Buttenfield. 1991. “Reconsidering Rules for Point-Feature Name Placement.” *Cartographica* 28(1): 10-27. doi: 10.3138/Y121-5621-8888-5306

Yamamoto, M., G. Camara, and L.A.N. Lorena. 2002. “Tabu Search Heuristic for Point-Feature Cartographic Label Placement.” *Geoinformatica* 6(1): 77-90. doi: 10.1023/A:1013720231747

Yoeli, P. 1972. “The Logic of Automated Map Lettering.” *Cartographic Journal* 9(2): 99-108. doi: 10.1179/000870472787352505

Zhang, Q. and L. Harrie. 2006. “Real-Time Map Labelling for Mobile Applications.” *Computers, Environment and Urban Systems* 30(6): 773-783. doi: 10.1016/j.compenvurbsys.2006.02.004

Zighed, J. A., R. Abdesselam, and A. Hadgu. 2012. “Topological Comparisons of Proximity Measures”, In *Advances in Knowledge Discovery and Data Mining Lecture Notes in Computer Science* 7301: 379-391. doi: 10.1007/978-3-642-30217-6_32

Zoraster, S. 1986. “Integer Programming Applied to the Map Label Placement Problem.” *Cartographica* 23(3): 16-27. doi: 10.3138/9258-63QL-3988-110H

Zoraster, S. 1997. “Practical Results using Simulated Annealing for Point Feature Label Placement.” *Cartography and Geographic Information Science* 24(4): 228-238. doi: 10.1559/152304097782439259

7 Improving Label Placement Quality by considering Basemap Detail with a Raster-Based Approach

Authors	Maxim A. Rylov and Andreas W. Reimer
Journal	*GeoInformatica*
History	Published (online): 10.07.2014 Accepted: 24.06.2014 Revised: 14.05.2014 Submitted: 20.01.2014
Reference	Rylov, M.A., and A.W. Reimer. 2014. “Improving Label Placement Quality by considering Basemap Detail with a Raster-Based Approach.” *GeoInformatica*. doi: 10.1007/s10707-014-0214-6

7 Improving Label Placement Quality by considering Basemap Detail with a Raster-Based Approach

Abstract

Topographic maps are arguably one of the most information-dense, yet intuitively usable, graphical artifacts produced by mankind. Cartography as science and practice has developed and collected a wealth of design principles and techniques to cope with the problems of high graphical density, especially for the case of label placement. Many of the more sophisticated techniques that take into account figure-ground relationships for lettering have not been fully operationalized until now. We present a novel generic quality evaluation model that allows full automation of refined techniques for improving map feature overlap, visual contrast and layer hierarchy. We present the objective function as a set of metrics corresponding to the design principles and provide exemplary parameterization via the set of experiments on global real-world datasets. The approach designed for labelling of point-like objects and can potentially be applied to linear and areal features. It has a low computational and memory requirement. Furthermore, it is conceivably applicable to annotate any kind of visualization beyond maps. The results of the conducted tests and comparison with a commercial labelling package illustrate the ability to produce highly legible and readable map lettering with our approach. Presented method heeds more cartographic design principles and is computationally less costly compared to commercially available methods.

Keywords: automated label placement, automated cartography, quality evaluation, image segmentation, GIS mapping

7.1 Introduction

Feature annotation is an important and a complex task in the process of spatial information representation and visualization. The task occurs in several disciplines like cartography (Yoeli, 1972; Imhof, 1975; Robinson et al., 1995), Geographic Information Systems (GIS) (Freeman, 1991), 3D modeling (Götzelmann et al., 2005; Lehmann and Döllner, 2012), and chart and graph drawing (Battista et al., 1994; Kakoulis and Tollis, 2003). Manual label placement is known to be a highly laborious and very time-intensive task. Nowadays, in the era of computers, there have been numerous and various research endeavors to automate the process of positioning names (see collection of papers maintained by Wolff and Strijk (2009)). Most of the previous research has dedicated to the problem of automated label placement in map lettering. The effectiveness and functionality of a map, as a communication medium, undoubtedly depends on how it is annotated. Clarity and legibility are two main objectives which the cartographer strives to achieve and which have direct influence on the perceptual and cognitive process used by map readers to search a certain name on the map and to determine its meaning (Noyes, 1980; Lloyd, 1997). Among other factors, map background, visual contrast (or visual clutter) and number of distractors highly affect visual search and reaction time (Brown, 1976; Phillips and Noyes, 1982; Lloyd, 1997). To relieve the density of a material, toponyms can be placed in regions with lower *graphic complexity* (Castner and Eastman, 1985), which allows reducing map complexity (Fairbairn, 2006). These regions are also attractive in the sense of higher type legibility (Phillips and Noyes, 1977).

In the literature, two different methods are used to evaluate the quality of label placement (van Dijk et al., 2002; Hong et al., 2005) in respect of overprinting of other geographic features. They are a *raster-based* (Doerschler and Freeman, 1992) and a *vector-based* method (Freeman and Ahn, 1984; Strijk and van Kreveld, 2002). The vector-based method generates its scoring for a given candidate position of a label by intersecting the label geometry with all potentially intersecting geometries of the background. Commercial packages like ESRI's Maplex Label Engine (2009) or MapText's Label-EZ use that technique and allow assigning preferences and weights to specific background feature classes. The vector-based approach is more appropriate for sparse maps with moderate feature density. It is more accurate in estimating the intersection of labels with background features, but meets insuperable difficulties in terms of computational complexity when the feature density of a map grows. The runtime of this method is $O((n+m)\log(n+m))$ (Strijk and van Kreveld, 2002), where n is the number of points to be labeled and m is the combinatorial complexity of the map features which can consist of a potentially enormous number of line segments. For geo-data that is not cartographically generalized, e.g., simplified, the problem is even more pronounced due to high vertex counts. The vector-based approach furthermore does not take into account the cartographic appearance of lines (e.g., width, stroke, cap, etc.) or polygons (e.g., hatching) directly without extra computations. The raster-based method assumes the map has already been rendered and looks at the content of the area covered by the label. Succinctly, the raster-based method (Doerschler and Freeman, 1992) is less geometrically accurate due to the discretization. At the same time it is robust to the density of map features and works with any data disregarding their degree of generalization. It does not require vector data as a data source, which allows it to be used for post-hoc labelling of graphics from external sources.

The problem of overprinting other non-textual map features by names received insufficient attention in the literature concerning automated label placement. One of the earliest attempts to use a rasterized map for identification of not occupied regions for making a label placement was made by Jones (1989). Jones proposed to prioritize pixel values according to the priority of the class of a background feature. A similar idea has been used by Harrie et al. (2004) and Zhang and Harrie (2006a). Their algorithm tries to position lettering so, that labels obscure cartographic data as little as possible. Stadler et al. (2006) adopted some ideas of morphological image processing to perform initial positioning of labels in their force-directed method.

This article focuses on defining an exhaustive quality measure for scoring potential label positions in terms of amount of clutter and overlapping other map features, lying in the map background. In our case the map background presents a raster image in which the map objects are already rendered. We define a measure which considers the extent, the shape and the importance of background features by extracting information from a raster image. In our approach we used the technique of *image segmentation* (Haralick and Shapiro, 1985; Pal and Pal, 1993) to perform analysis of a pixel-based image and methods for measuring colour similarity and for measuring *the information content* in maps (Li and Huang, 2002; Harrie and Stigmar, 2010). The distinguishing peculiarities of our measure are the following:

- Appropriate for any feature type: point (Carstensen, 1987; Rylov and Reimer, 2014a), line (Edmondson et al., 1996; Wolff et al., 2001) and polygon (van Roessel, 1989; Barrault, 2001; Rylov and Reimer, 2014b).
- Any nature of the background: diverse raster data sources, maps with terrain representation (Imhof, 1982/2007; Jenny and Hurni, 2006), Web Map Service (WMS; OGC, 2006) or Web Map Tile Service (WMTS; OGC, 2010; García et al., 2012) layers.
- Applicable to data of any scale, i.e. with any degree of generalization without impact on the runtime.
- Low computational complexities and memory requirements.
- Capability to be used for annotating objects in interactive 3D maps (Lehmann and Döllner, 2012).

The proposed measure can be considered as an additional component of a generic quality evaluation function (van Dijk et al., 2002) that measures how good a certain labelling algorithm performs its task. As a rule, this function is employed by a *mathematical optimization* algorithm for solving automated label placement problem (Edmondson et al., 1996; Rylov and Reimer, 2014a).

This paper begins with giving some principles of automated label placement and how the presented approach supplements existing methods (Section 7.2). Further, we continue with writing out some suited design principles in the form of cartographic guidelines and describing the quality measure in sufficient detail (Section 7.3). Next, we present the results of our empirical experiments on real-world datasets, provide exemplary parameterization and show some sample maps that were annotated using our measure (Section

7.4). The results illustrate the advantages and abilities of the proposed measure. Finally, we conclude with an analysis of the measure and provide thoughts about possible use cases (Section 7.5).

7.2 Principles of Automated Label Placement

Many various compelling techniques have been proposed over last four decades to solve the problem of automated label placement. Among them are *rule-based "expert" systems* with exhaustive search (Yoeli, 1972; Hirsch, 1982; Ahn and Freeman, 1984), integer programming (Zoraster, 1986), artificial intelligence procedures (Verner et al., 1997; Schreyer and Raidl, 2002) or metaheuristic search methods (Christensen et al., 1995; Yamamoto et al., 2002). Most researchers (e.g., Cromley, 1985; Zoraster, 1986) consider label placement problem as a problem of combinatorial optimization (Schrijver, 2003). Thus, this kind of a problem can be solved using mathematical optimization algorithms (Edmondson et al., 1996), which require two components to be defined: a discrete search space and an objective function. In the context of label placement problem a search space consists of elements that represent candidate label positions. The purpose of an objective (quality) function is to measure the goodness of a certain label position (an element of a search space) with respect to other labels and to the map content as a whole. An objective function gives a numerical score that indicates the quality of labelling. In other words, the quality function aids to imitate the process, employed by a cartographer, of finding a compromise between diverse informal and contradicting principles of good label arrangement. As map labelling quality depends on many cartographic criteria then the general form of an objective function is normally defined as a weighted sum of single metrics (van Dijk et al., 2002; Zhang and Harrie, 2006b; Rylov and Reimer, 2014a):

$$\begin{gathered} F(L) = \sum_{l \in L} \big(w_1 \cdot f_{\text{priority}}(l) + w_2 \cdot f_{\text{aesthetics}}(l) + \\ + w_3 \cdot f_{\text{association}}(l) + w_4 \cdot f_{\text{label-visibility}}(l) + w_5 \cdot f_{\text{feat-visibility}}(l)\big) \end{gathered} \tag{7.1}$$

where $L = (l_1, \ldots, l_n)$ is a set of n labels on the map, w_i, $i = 1,2,\ldots,5$ are the weights. The partial quality functions $f_*(l)$ are:

- The function $f_{\text{priority}}(l)$ measures the importance, classification and hierarchy of a labeled object.
- The aesthetic quality of a label is denoted as $f_{\text{aesthetics}}(l)$ and evaluates the quality of the position and the shape of a label with respect to geometry of the feature it annotates.
- The degree of association between a particular feature and its label is measured by the $f_{\text{association}}(l)$.

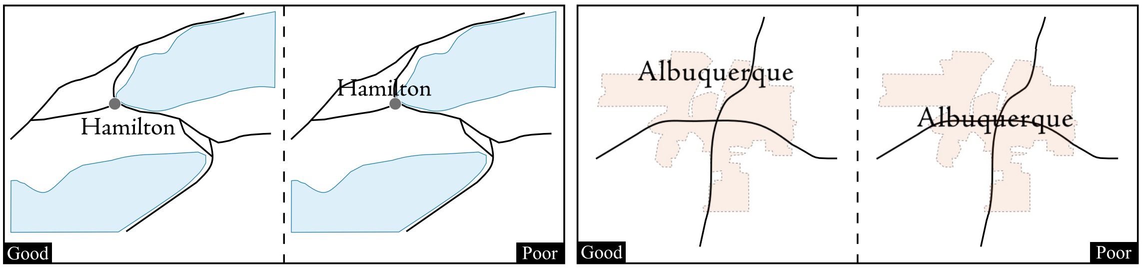
- The $f_{\text{label-visibility}}(l)$ represents how well a label is visible by taking into account other features and labels on the map.
- The metric $f_{\text{feat-visibility}}(l)$ represents a measure of quantifying how well a feature is visible on the map with respect to other features and labels.

An extensive study, which contains about sixty criteria on the issue of measuring the quality of label placement for any type of designation, was done by van Dijk et al. (2002).

In this paper we consider a refined measure which substitutes metrics $f_{\text{feat-visibility}}(l)$ and $f_{\text{label-visibility}}(l)$ in (7.1). The measure quantifies amount of the map information that can be hidden by a label and the degree of visual contrast between a label and other graphical elements when the map background is presented as a raster image. The devised measure relies on the information derived from the map background after applying an image segmentation algorithm (Haralick and Shapiro, 1985; Pal and Pal, 1993).

7.3 Measures of Feature Overprinting and Type Legibility

One of the goals of automated label placement is to formalize and quantify cartographic principles, which are given in descriptive form, into a set of numerical measures. The ability of the algorithm to find a good type placement highly depends on how good those measures perform their task. In this section we introduce some quality metrics which can measure the degree of label-feature overprinting and visual contrast between them. In other words, the metrics allow evaluating and consequently minimizing distortion of map features that has been effected due to lettering. They also allow enhancing map legibility. The devised metrics are based on widely used cartographic precepts.



(a) (b)

Figure 7.1. Illustrative examples of good and poor name positioning for the above-mentioned guidelines.

7.3.1 Cartographic Principles

Cartographers have well-established and descriptive rules and postulates, which explain the principles of map lettering and serve as a guide for producing good type placement on maps (Imhof, 1975; Wood, 2000; Brewer, 2005). We studied sufficiently broad and comprehensive cartographic guidelines that refer to overlapping and clutter of other map symbols (e.g., lines, hachures, shaded hills, etc.) by names. These certain guidelines are the following:

G1. Move the names away from positions where they partially overlap or even totally conceal other symbols. Locating names in empty spaces is preferable (Figure 7.1a).

G2. Avoid interfering or overprinting a geographic feature which is running the length of the name (Figure 7.1b)

G3. For legibility, the name should be placed so as to minimize visual contrast of other names and space around them (Figure 7.2).

G4. Labels should take into account the nature and the importance of the graphical features in the layout. Overlapping or concealment of different feature classes (e.g., roads, lakes, rivers, administrative boundaries) by types should be treated differently. Positioning names on top of less important features is preferable.

In the following subsections we present and describe in detail four metrics (Section 7.3.3-7.3.6), which correspond to the cartographic guidelines given above. We conclude with a single quality evaluation function that incorporates all four metrics at once (Section 7.3.7).

7.3.2 Preliminary Definitions

This subsection gives some basic definitions which help us to provide the description of the proposed metrics.

As mentioned above, our measure depends on a raster image which represents non-textual map features. In order to make further work with raster data (pixels) easier, we chose the approach of image segmentation (Haralick and Shapiro, 1985; Shapiro and Stockman, 2001). Such representation of an image facilitates efficient and meaningful analysis of its properties. Besides, this method is extensively used for feature extraction and recognition from digital cartographic documents (Leyk and Boesch, 2010). In our approach image segmentation helps to merge the map background regions which have similar colours. Image segmentation consists of the partitioning of an image into homogeneous regions (clusters). We note that our approach assumes clusters based on colour space only, disregarding spatial proximity. Maps in raster form can contain a large

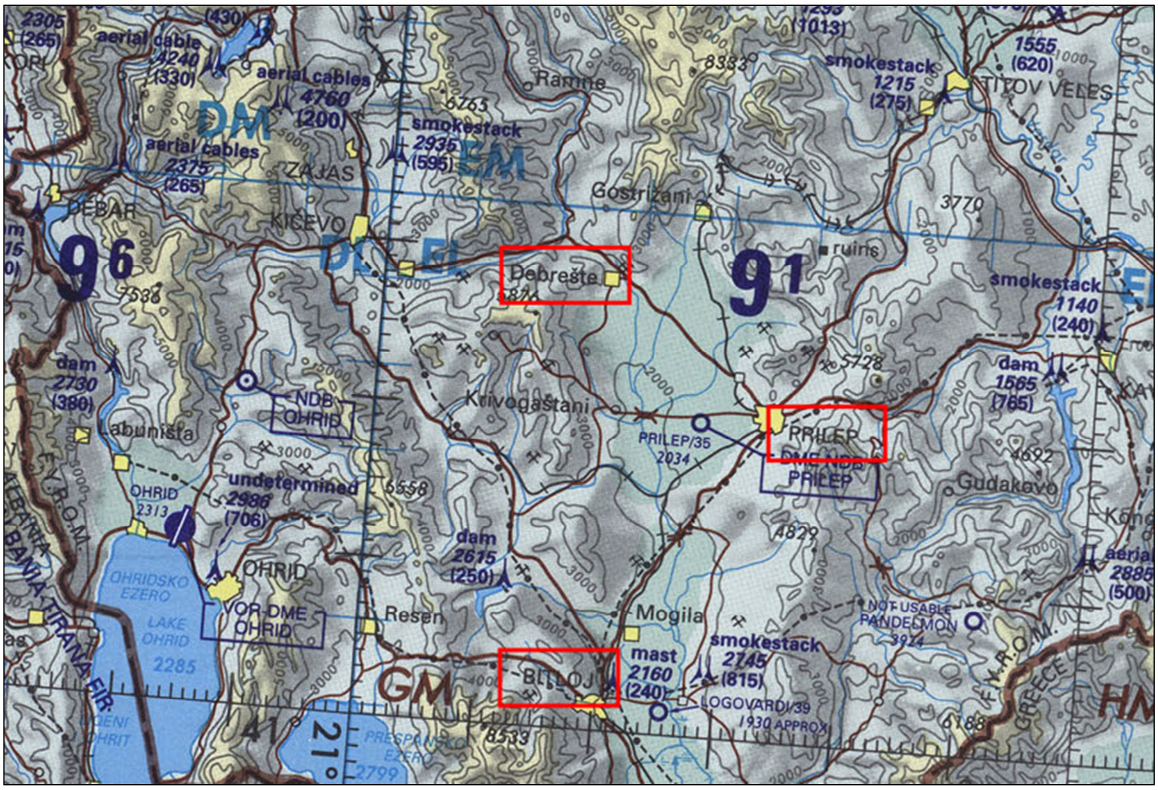


Figure 7.2. An example map lacking of visual contrast between labels (e.g., Prilep city, Bitola city, Debrešte village) and map background. SOURCE: Defence Geographic and Imagery Intelligence Agency (UK) (2000), © UK MOD Crown Copyright, 2014.

number of unique colours that can be governed by shaded relief (Imhof, 1982/2007; Jenny and Hurni, 2006) or bathymetry, gradient fills of polygons or colour gradations produced by the antialiasing technique (Forrest, 1991). In this case, the clustering is crucial for further analysis as a data reduction step. Next, we omit the description of any existing image segmentation method and assume that one of them has already been chosen.

The required input for our measure is a digital image, the number of clusters in the segmented image and a set of axis-aligned rectangles that represent the boundaries of the characters in a name, where each character can be a letter or other graphic symbol. In simple terms, the raster pixels are grouped and succinctly indexed by colour. We further require the boxes around the individual letters to line up with the raster grid. The following paragraph expresses this process more formally.

Let us define $I = \{1, \ldots, W\} \times \{1, \ldots, H\}$, where W and H are the dimensions of the input image, $I \subset \mathbb{Z}^2$. Each element $p \in I$ is a pixel that has its colour, denoted as C_p.We also denote the number of clusters in the segmented image as M. We assume that an image segmentation algorithm φ (Haralick and Shapiro, 1985) has been applied to the

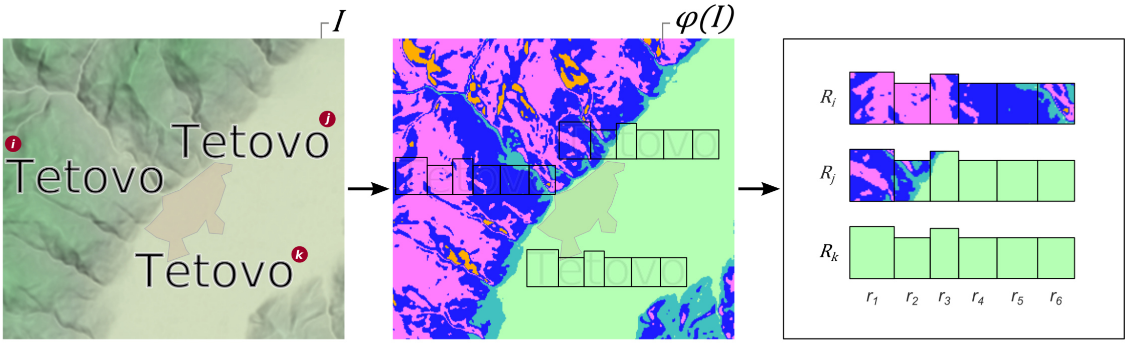
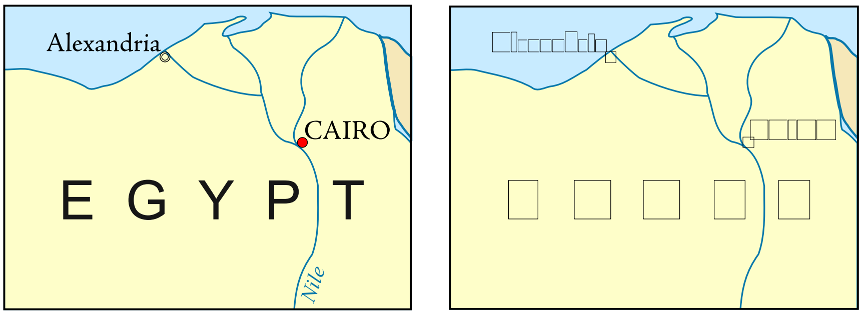


Figure 7.3. Workflow of the method. The input image I with shaded relief is transformed into a segmented image (M=4). Next, map background information of the segmented image is assigned to each letter of the potential labels. Note that the 'urban area' of Tetovo is imaginary and has been provided for the sake of explanation.

input image I. The algorithm φ was able to assign an index of a cluster to each pixel $p \in$ el $p \in I$. Thus, we can denote this transformation as $\varphi(I) = \{S^1, S^2, \dots, S^M\}$, where $S^m, m = 1, \dots, M$ are the clusters. Hence, each pixel $p \in I$ has the associated cluster index s_p. Further, let us assume that the name of a label $l \in L$ consists of K_l characters. Then, the set of axis-aligned rectangles that bound its characters we denote as $R_l = \{r_1, \dots, r_{K_l}\}$, where $r_i,\ i = 1, \dots, K_l$ (see Figure 7.3). In our implementation of the metrics, we demand that beforehand the coordinates and size of $r_i$are rounded to the pixel coordinates (see Figure 7.7). In order to shorten further mathematical manipulations, we denote all pixels that lie within a certain rectangle r_i as D_{r_i}. We also define the area of an image covered by a label l as $A(R_l) = \sum_{i=1}^{K_l} a(r_i)$, where the function a returns the area of a rectangle r_i, or, in other words, the number of pixels. Note that the pixel size is only considered in the procedure of rounding the character bounds r_i (see also Section 7.3.9).



(a)

(b)

Figure 7.4. A sketchy representation of axis-aligned rectangles (b) which bounds characters of a label (a).

The rectangle-based representation of type letters also supports the cartographic technique known as letter-spacing which is a powerful and widely used design element. Sometimes it is called type spacing, character spacing or tracking in typography. Letter-spacing gives more freedom in making the names and the non-textual features less obscure (e.g., see Figure 7.4, the Nile River goes between two letters).

7.3.3 Measure of Background Homogeneity

A good design technique is to place labels in areas where other features are less dense. The purpose of this sub-section is to define a metric that can measure *background homogeneity* (see G1). In image processing, homogeneity expresses how similar certain elements (pixels) of the image are. Homogeneity has diverse definitions and measures, which can be found in the literature (Pal and Pal, 1987; Jurio et al., 2013). Pursuing our needs, we consider homogeneity only for the pixels that are covered by the characters of a bel $l \in L$, namely by the set of given rectangles in R_l. Hence, we can use the concept of local homogeneity which we define as a value that represents a cluster with maximum number of elements bounded by R_l. Let us define a function that calculates the number of elements of a cluster m within R_l as follows:

$$N_{\mathrm{cl}}(R_l, m) = \sum_{i=1}^{K_l} \sum_{q \in D_{r_i}} B(s_q, m) \tag{7.2}$$

$s_q \in \{1, \dots, M\}$ is a cluster index of a pixel q. The function $B(d, e)$ defined as:

$$B(d, e) = \begin{cases} 1, & d = e \\ 0, & d \neq e \end{cases} \tag{7.3}$$

where d, e are the cluster indices.

Hence, using (7.2), the background homogeneity metric can be written as:

$$Q_{\mathrm{BH}}(R_l) = \frac{\max\limits_{m \in M} N_{\mathrm{cl}}(R_l, m)}{A(R_l)}. \tag{7.4}$$

The function $Q_{\mathrm{BH}}(R_l)$ can be interpreted as the percentage of the area inside R_l covered by the cluster with maximum number of elements (pixels) in R_l. Function (7.4) returns a value in the interval [0,1]. This metric is designed to yield a value of 1.0 when all elements within R_l belongs to one cluster, i.e. the region of the map background under a label l is homogeneous. An application of metric (7.4) in a comprehensive multi-criteria model for point-feature label placement can be found in the work by Rylov and Reimer (2014a).

7.3.4 Measure of Spatial Distribution

The shortcoming of the measure presented in the previous sub-section is that it does not consider the *spatial distribution* of the clusters over the rectangles. This fact can be revealed by a simple example. In Figure 7.5 two different distributions of a certain cluster (grey pixels) are shown. In the examples (a) and (b) the cluster has exactly the same number of elements in R_l. Thus, according to the definition of $Q_{\mathrm{BH}}(R_l)$, this metric returns the same score value. However, as it can be seen from Figure 7.5, the distributions of cluster elements within the rectangles are absolutely different. In example (a) all elements of the cluster are mostly located in the first rectangle. But example (b) represents the case when the elements of the cluster are spread over the rectangles, in other words, along the name. This fact formally contradicts G2. Therefore, a new metric that considers the spatial distribution of background features is needed. An intuitive answer would be to simply count the number of rectangles that contain elements of a given cluster. There are many cases, though, where cluster elements are present in all rectangles and a comparison between their respective distributions is still needed.

To go one step further with the intuitive answer by weighting the occurrence of overlaps by the percentage of elements in regard to the whole is very close to the concept of "entropy" for quantification of information (Shannon and Weaver, 1964). Sukhov (1967, 1970) has borrowed the idea of the concept and applied it for measuring the information content in maps. Later, his quantitative measure was revised by Li and Huang (2002) to consider the spatial distribution of the map objects. Their normalized entropy (see also Harrie and Stigmar, 2010) is defined as:

$$H = \frac{\sum_{j=1}^{n} P_j \cdot \log P_j}{\log\left(\frac{1}{n}\right)} \tag{7.5}$$

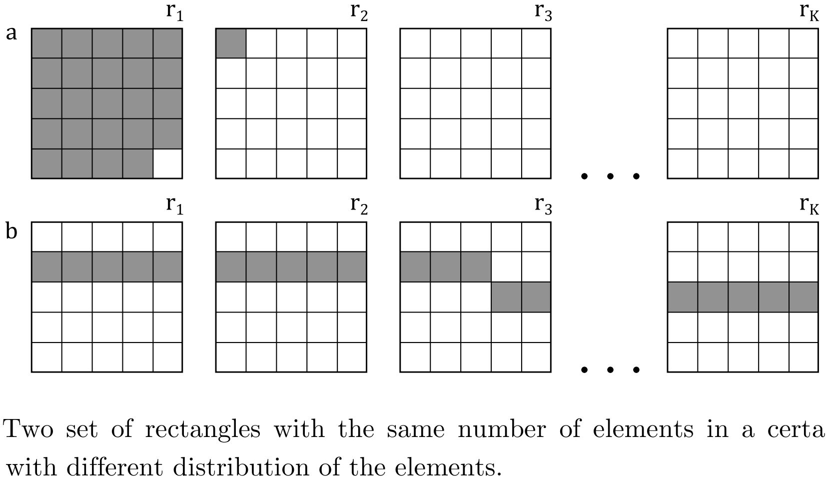


Figure 7.5. Two set of rectangles with the same number of elements in a certain cluster, but with different distribution of the elements.

where P_j is the probability which is calculated as the ratio between the area of Voronoi cell of an object j and the total area of the map, n is the number of objects.

In pure form the measure defined in (7.5) is not appropriate for using in our approach. Therefore, we devise to accommodate it to meet our needs. We assume that the total entropy for spatial distribution of the clusters within R_l is calculated as a sum of the entropies of particular clusters:

$$H(R_l) = \sum_{m=1}^{M} \frac{H_{\mathrm{cl}}(R_l, m)}{M} \tag{7.6}$$

where $H_{\mathrm{cl}}(R_l, m)$ is the entropy of the cluster S^m within R_l. The interpretation of the entropy in our case is the following: for the same number of cluster elements, the entropy will be larger if the elements are more evenly distributed over the rectangles of R_l. Now let us define the probability of the cluster S^m within r_i as:

$$P_i(R_l, m) = \frac{\sum_{q \in D_{r_i}} B(s_q, m)}{A(R_l)} \tag{7.7}$$

where $P_i(R_l, m)$ is the probability for the cluster S^m in the ith rectangle, function B is defined in (7.3). Then, the entropy of the cluster S^m can be calculated, according to (7.5), as follows:

$$H_{\mathrm{cl}}(R_l, m) = \begin{cases} \dfrac{\sum_{i=1}^{K_l} P_i(R_l, m) \cdot \log P_i(R_l, m)}{\log\left(\dfrac{1}{N_{\mathrm{cl}}(R_l, m)}\right)}, & N_{\mathrm{cl}}(R_l, m) > 1 \\ 0, & \mathit{otherwise} \end{cases} \tag{7.8}$$

where $N_{\mathrm{cl}}(R_l, m)$ is a function, defined in (7.2), that returns the number of elements, which belong to the cluster S^m in all rectangles of R_l.

In order to meet the requirements of G2 we need to construct the spatial distribution metric in such a way that it returns values that are close to 0.0 when the clusters within R_l are evenly distributed. Conversely, the metric should return a value of 1.0 when the clusters are not evenly distributed. Hence, the final metric can take the form of:

$$Q_{\mathrm{SD}}(R_l) = 1 - H(R_l) \tag{7.9}$$

where $H(R_l)$ is defined in (7.6). The function $Q_{\mathrm{SD}}(R_l)$ takes values in the interval [0,1].

7.3.5 Measure of Background Feature Priority

Toponyms may overlap non-textual background elements of different classes (e.g., land, roads, seas, woods) which of course have distinct importance (G4). Jones (1989) proposed to use an overlapping priority for different feature classes, i.e. each colour of the background has a priority. As a label covers set of pixels, we can conclude that the best label

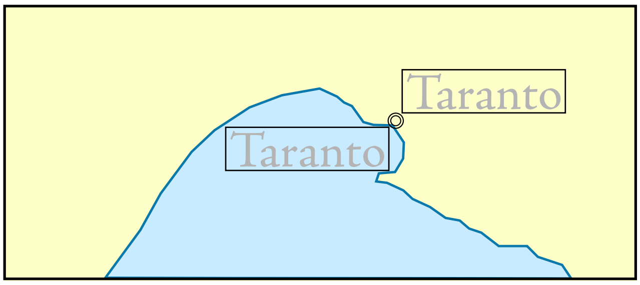


Figure 7.6. An example of possible label positions which have the same value of Q_{BH} and Q_{SD}.

position is in which the sum of priorities of covered pixels is minimal. Hence, using aforesaid notations we can construct the next metric as:

$$Q_{\mathrm{FP}}(R_l) = 1 - \frac{\sum_{i=1}^{K_l} \sum_{q \in D_{r_i}} P(s_q)}{A(R_l)} \tag{7.10}$$

where $P(s_q)$ is a function that returns the priority of a background pixel q. Function $P(s_q)$ should return normalized values, in order to take in (7.10) the values in the range [0,1]. Metric (7.10) also helps to distinguish and refine potential label positions that are considered by the metrics Q_{BH} and Q_{SD} as equal (see Figure 7.6).

7.3.6 Measure of Visual Contrast

In this sub-section we consider a metric that measures *visual contrast* (G3) between a label and the map background in terms of their colour similarity. Earlier works by Williams (1967) and by Phillips and Noyes (1982) showed that clutter mostly comes from symbols of similar colours. It was proved through the set of experiments that the effect of lacking visual contrast decreases the performance of map reading. Wood (1994) experimentally determined that brightness difference between figure and ground is highly important for the tasks of estimation and comparison of map areas. Furthermore, Swiss cartographer Eduard Imhof asserted:

> *"One should always combine those elements which are as different or as contrasting as possible, both in the information which they contain and in their design characteristics: in other words, those symbols which supplement each other well in what they represent and are graphically compatible."*
>
> Eduard Imhof, 1982, p. 334

These facts lead us to the necessity of taking into account contrast difference in the quantification of feature overlapping, since this may positively affect the type legibility and visual search time as a whole.

We base our metric on the difference between two colours. It is obvious that the difference between the colours, as they are perceived, determines the figure-ground relationship. The colour space (e.g., RGB, CMYK, HSV, etc.) plays a crucial role in measuring the colour-difference. Since we are interested in measuring visual perception of the information, the colour space should be designed in a way of good approximation of human vision. One of such colour spaces is *CIE Lab* (CIELAB) space (1978) which was standardized by the French Commission Internationale de l'Eclairage (International Commission on Illumination). A colour in CIELAB is expressed through three components $L^*, \mathrm{a}^*, \mathrm{b}^*$, where L^* defines lightness, a^* and b^* denote red/green and yellow/blue values respectively. The distance between two colour values in CIELAB is defined as:

$$\Delta E(x,y) = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2} \tag{7.11}$$

where $\Delta L^*, \Delta a^*, \Delta b^*$ are the differences between corresponding components.

To compute the difference of colours between a label and the background, we need to calculate the difference between the colour of the label and colour of each pixel in R_l. Hence, we can define visual contrast metric as follows:

$$Q_{\mathrm{VC}}(R_l) = \frac{\sum_{i=1}^{K_l} \sum_{q \in D_{r_i}} \Delta E(C_q, C_{\mathrm{t}})}{100 \cdot A(R_l)} \tag{7.12}$$

where C_q, C_{t} are the colours of a background pixel q and text colour respectively. The function $Q_{\mathrm{VC}}(R_l)$ is normalized and its values fall in the range [0,1]. The value 100 in the denominator represents the difference of the lightness component L^* of black and white colours.

It should be noted that the problem of colour perception in cartography is much more profound than the CIELAB-space distances suggest. On a fundamental level, this is caused by the inverse optics problem (Wojtach, 2009; Reimer, 2011). For our label placement problem, the CIELAB approximation works well as a proof of concept. For more sophisticated techniques trying to address simultaneous contrast and perceptual distances, we point the reader to the works like Lee et al. (2013) for pixel-based techniques and Purves and Lotto (2010) for more holistic approaches.

7.3.7 Aggregated Measure

In the previous section, four metrics have been proposed. They quantify the cartographic guidelines given in Section 7.3.1. These metrics, which are defined in (7.4), (7.9), (7.10) and (7.12), can be conflated into one single measure as follows:

$$Q(l) = \alpha_{\mathrm{BH}} \cdot Q_{\mathrm{BH}}(R_l) + \alpha_{\mathrm{SD}} \cdot Q_{\mathrm{SD}}(R_l) + \alpha_{\mathrm{FP}} \cdot Q_{\mathrm{FP}}(R_l) + \alpha_{\mathrm{VC}} \cdot Q_{\mathrm{VC}}(R_l) \tag{7.13}$$

where R_l is the bounds of a label $l \in L$ and $\alpha_{\mathrm{BH}}, \alpha_{\mathrm{SD}}, \alpha_{\mathrm{FP}}, \alpha_{\mathrm{VC}}$ are weighing parameters which should sum up to 1 in order to yield values of $Q(l)$ in the interval [0,1]. The higher the value of the function the more preferable a label position is. Equation (7.13) is a counterpart of functions $f_{\text{label-visibility}}(l)$ and $f_{\text{feat-visibility}}(l)$ in equation (7.1). For an exhaustive study on the issue of construction of quality functions please refer to van Dijk et al. (2002). The approach of adjustable weights affords the opportunity to prefer one cartographic guideline over another. The adjustment of the weights should be undertaken by the experienced user with care. The aim of the adjustments is to find optimal set of weights, which produces the most readable, legible and aesthetically plausible type placement. Section 7.4.3 includes a detailed discussion of possible values of the weights and their influence on the resulting labelling.

7.3.8 Computational Complexity

In this section, we discuss the factors influencing the overall computational complexity of the proposed model. It should be noted that the computation of our composite measure is performed only once, namely, after all potential label positions have been generated and before the position selection procedure (e.g., greedy, discrete gradient descent or simulated annealing algorithm, see Edmondson et al., 1996) is applied.

The time complexity is determined by two main stages that constitutes the two parts of our approach. They are the image segmentation algorithm and the computation of the quality score for each label using function (7.13). Let N be the number of pixels in the input image I, and let M be the number of the required clusters. Denote by V the number of pixels covered by n labels. Then, the time complexity of the model can be written in the general form as $O(F(N,M)+nM+V)$, where $F(N,M)$ is a function that defines the number of operations for the image segmentation algorithm. The second term of the total complexity $O(nM+V)$ represents the runtime which is needed to perform the labelling quality evaluation using our measure. Taking into account that M is fixed and has small values in practice, V directly depends on n and on the font size of the text, the second term can be computed in linear time $O(n)$. Note that the first term $O(F(N,M))$ depends on the type of segmentation technique and can vary greatly. For example, the seeded region growing method (Adams and Bischof, 1994) has the time complexity of $O((M+\log \mathrm{N})N)$. The K-means clustering algorithm requires $O(NMr)$ execution time, where r is the number of iterations taken by the algorithm to converge (Jain et al., 1999). Furthermore, the hierarchical agglomerative algorithm is much slower and requires $O(N^2 \log N)$ (Jain et al., 1999).

We summarize that our measure can perform its task very rapidly in comparison to the whole runtime consumed by other stages of a labelling algorithm (see details in Section 7.4.3) when it deals with the moderate number of labels and a fast image segmentation algorithm.

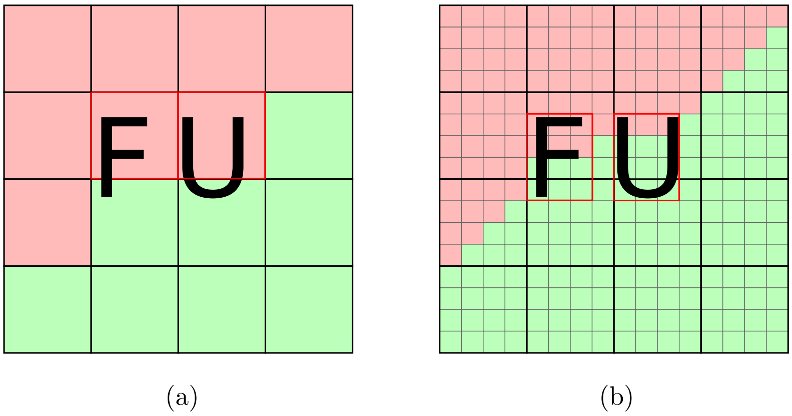


Figure 7.7. An example of letter bounds alignment to the pixel grid with different pixel resolutions. The resolution of image (b) is four times higher than in image (a).

7.3.9 Image Resolution

In practical situations, a raster image that represents map background can have an origin different from the labels to be rendered. It follows that their raster resolution might be different. The raster resolution plays an important role in the accuracy of the presented method. The two factors that have an influence on the approach can be defined as:

- Rounding error in the alignment of the letter bounds to the pixel coordinates (see red rectangles in Figure 7.7a).
- A digital image with lower resolution providing less information for the estimation of map content covered by labels (see Figure 7.7).

In general, we can conclude that the higher resolution of the input raster is the better our measure accomplishes its task. At the same time the runtime is increased, as it is needed to operate with larger number of pixels both in image segmentation and quality evaluation steps. To obtain plausible results the input image for our measure should obviously have the resolution equal to or greater than the resolution of the text that is being positioned.

7.4 Implementation and Experiments

In this section we present the results of some experimental tests that were conducted. The results show the evidence of applicability of our measure in labelling of the sample maps which are based on real-world geographic datasets. Moreover, we try to study the influence of each metric (see Section 7.3.3-7.3.6) on the resulting type placement.

7.4.1 Implementation

We have implemented a version of the measure presented in this paper within a framework for producing maps on desktop or publishing them to the web. This framework is written in C# and called *MapSurfer.NET*. The experiments have been performed on a machine with an Intel® Core™ i5-2500 CPU @ 3.30 GHz running Windows 7 Professional x64 with 8GB installed memory. The runtime execution environment of our test application was .NET Framework 4.5 (x64).

For the experiments we have chosen the *octree* quantization algorithm (Gervautz and Purgathofer, 1988) to perform image segmentation of an image that represents the map background. This algorithm is very simple in implementation and provides the best compromise of the *performance* and the quality in comparison to other more sophisticated methods (e.g., Leyk and Boesch, 2010). The number of clusters (colours) was set to 8. The choice of the image segmentation algorithm and the number of clusters is, of course, arbitrary. But, note that they should depend on the nature and the content of the map background. In order to properly evaluate the positions of the labels that partially lie inside the map viewport, we use a raster image with enlarged extent according to the label coverage. In our implementation, the map background was fetched from a WMTS service (OGC, 2010; García et al., 2012) that provided tiles which are combined into a single image covering the whole region of interest.

In our experiments we applied the proposed measure only to point features which were labeled using the fixed position model, namely the 8-positioned model (Figure 7.8). Position selection procedure in the type placement algorithm was performed using simulated annealing algorithm (Kirkpatrick et al., 1983) proposed by Christensen et al. (1995). To evaluate the quality of labelling, we used a modified version of the objective function (7.1). We substituted the metrics $f_{\text{label-visibility}}(l)$ and $f_{\text{feat-visibility}}(l)$ for the measure (7.13). The final quality function in our tests had the form:

$$F(L) = \sum_{l \in L} \left(w_1 \cdot f_{\text{priority}}(l) + w_2 \cdot f_{\text{aesthetics}}(l) + w_3 \cdot f_{\text{assoc.}}(l) + w_4 \cdot Q(l) \right) \tag{7.14}$$

The description of other metrics in (7.14) is suppressed in this paper, since they are given in sufficient details by Rylov and Reimer (2014a).

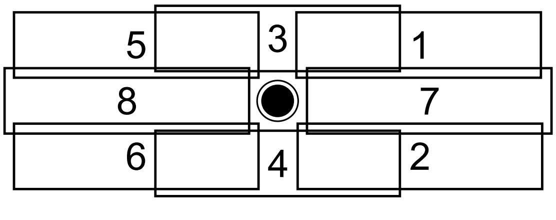


Figure 7.8. Imhof's (1975) model for positional prioritization of point-feature labelling.

7.4.2 Dataset Information

Two datasets have been used in the experiments. The first dataset contains point, linear and areal features that have been extracted from the dataset which represents *Volunteered Geographic Information* (VGI; Goodchild, 2007), more precisely the data of the *OpenStreetMap* (OSM) (Haklay and Weber, 2008; Ramm et al., 2010). The second dataset represents *digital elevation model* that was collected using Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER GDEM; 2011).

7.4.3 Experimental Results

For the first set of tests, we have chosen a map region which represents the island of Sicily. The image of the map background contains the road network and water bodies. In order to determine the general contribution of the proposed measure, we first produced the label placement without the measure, i.e. in equation (7.14) $w_4 = 0$. The resulting map is depicted in Figure 7.9. From that figure we clearly see that there are many labels which partially conceal or entirely obliterate background features (see labels marked with red frames). For example, the city of Caltanissetta completely hides a segment of the road network. Another demonstrative example is a label of Randazzo town where the letters cover the whole length of a river (Fiume Alcantara).

Next, we carried out an experiment by using our measure with $w_4 = 0.4$ and the parameter set № 1 from Table 7.1. Thus, we incorporate the preference of background homogeneity and absolutely neglect the influence of contrast differences by setting $\alpha_{\mathrm{VC}} = 0.0$. For the metric Q_{FP} (Section 7.3.5) we set the lowest priority to the colour of the sea while other colours are equally important. Figure 7.10 illustrates the results of this experiment. It can be seen that the resulting map became much more clear and legible in terms of distortion of other map contents. In other words, the labels overprint the geographic symbols (e.g., roads, rivers) to a lesser extent. However, some labels are located in positions where the roads running the length of the name (e.g., the city of Caltanissetta and town of Palagonia, see Figure 7.10). Next, we tried to improve the lettering for above-mentioned places by using the metric of spatial distribution (Section 7.3.4) with the higher value of α_{SD}. We ran the label placement algorithm with the parameter set № 2. The result of this experiment is shown in Figure 7.11. Now we can observe that the

Table 7.1. Parameter sets of the experiments.

№	w_4	α_{BH}	α_{SD}	α_{FP}	α_{VC}	Mask content
1	0.4	0.85	0.1	0.05	0.0	roads and water bodies
2	0.4	0.7	0.25	0.05	0.0	roads and water bodies
3	0.2	0.7	0.3	0.0	0.0	roads and water bodies
	0.2	0.3	0.2	0.0	0.5	shaded relief

Figure 7.9. A map of Sicily annotated without the measure. Projection: spherical Mercator (EPSG:3857). Data source: OpenStreetMap project (2013).

names of places Caltanissetta and Palagonia are located in different positions, where they less interfere with the road network, while other labels remained mostly on the same positions. As Figures 7.10 and 7.11 show, the behavior and the influence of the two tested metrics Q_{BH} and Q_{SD} on the resulting labelling is rather subtle and requires a careful selection of the weights α_{BH}, α_{SD} in (7.13).

We provide the runtime measurements of the last experiment, to give an impression on the computational costs of the presented method. The input data for the tested region contain 120 named locations, resulting in 960 candidate label positions. Taking into account the set of quality measures (7.14), the simulated annealing produced 108 label placements in 2.45 seconds. Scoring all candidates with our proposed measure (7.13) took 179ms (83ms for image segmentation). This is equal to 7% of the overall time spent on labelling. Note that in our implementation we used only one core of the processor. Hence, the performance can be improved by exploiting multiple CPU cores for scoring labels with (7.13) or a GPU for image segmentation.

In order to compare the effectiveness of the presented approach with the vector-based method, we prepared a map in ArcMap 10.1 using the same dataset and performed label placement using Maplex (2009) (Figure 7.12). We configured Maplex to avoid overprint-

Figure 7.10. The labelling using the measure with the preference for background homogeneity. Projection: spherical Mercator (EPSG:3857). Data source: OpenStreetMap project (2013).

ing roads and water bodies. The presented map consists of 12422 individual background features. Maplex uses a version of the *sliding* model (Strijk and van Kreveld, 2002; van Kreveld et al., 1999) by operating with 96 candidate positions for each point-feature, yielding a total of 11520 candidate positions. Succinctly, Maplex was able to place 4 more labels (see Table 7.2) than the simulated annealing algorithm equipped with our raster-based measure. This is a highly-expected result. Strijk and van Kreveld (2002) already reported that the simple implementation of the sliding model outperforms simulated annealing by up to 10% in number of labels located. It is important to mention that Maplex's label placement has 4 label pairs in which labels overlap with nearby point symbols (see red circles in Figure 7.12: Giarre and Riposto, Misterbianco and Motta Sant'Anastasia, Carlentini and Lentini, Erice and Valderice). Maplex slightly outperforms our algorithm in the number of labels without any feature-overlap (35 vs. 32, Figures 7.12 and 7.11, Table 7.2), when considering the 105 labels common to both results. This is most likely a function of the greater degrees of freedom for the sliding model. The label placement without using measure (7.13) is poor and awkward due too many label-feature overlaps (Figure 7.9), markedly decreasing legibility and readability Next, we

Figure 7.11. The labelling using the measure with the preference for background homogeneity and spatial distribution. Projection: spherical Mercator (EPSG:3857). Data source: OpenStreetMap project (2013).

Table 7.2. Lettering placement characteristics produced by different labelling algorithms.

	Labelling without measure (7.13)	**Labelling using measure (7.13), set №2**	**Maplex**
Candidate positions	960	960	11520
Labels placed [number of label overlaps]	107 [0]	108 [0]	112 [4]
Labels without feature-overlap	$\frac{19}{103}$ [18.44%]	$\frac{32}{105}$ [30.48%]	$\frac{35}{105}$ [33.33%]
Preferred positions among the labels with feature-overlap		$\frac{30}{67}$ [44.78%]	$\frac{4}{67}$ [5.97%]

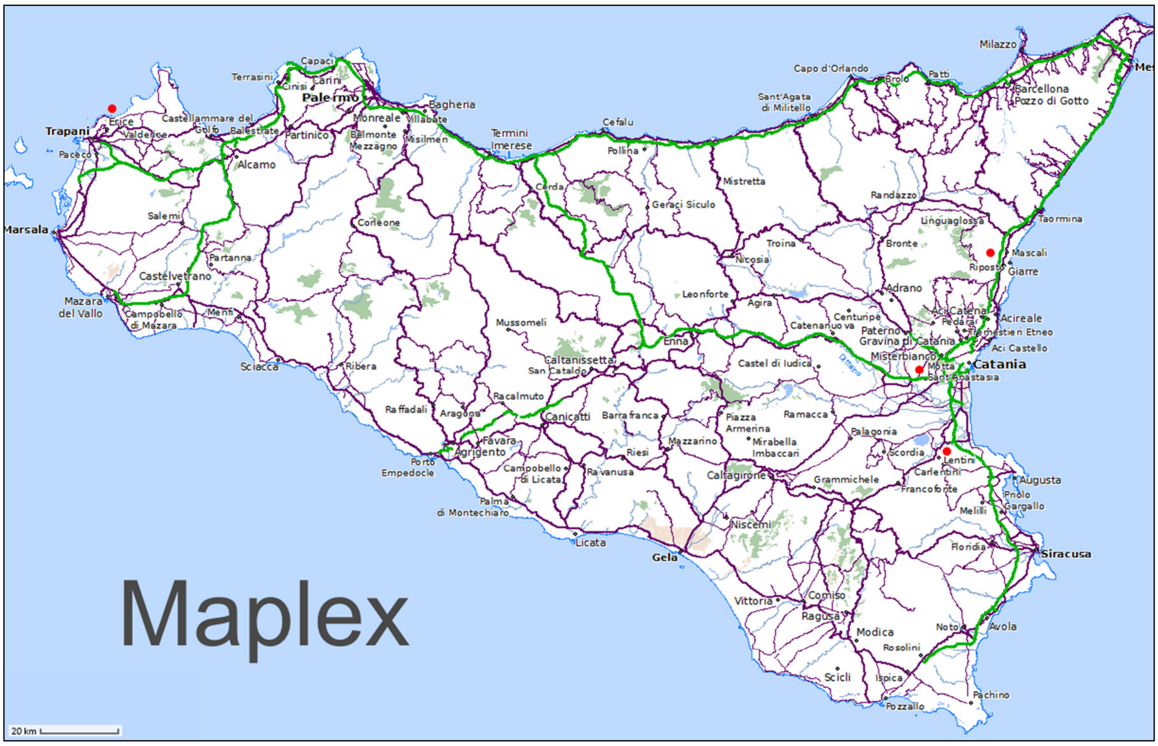


Figure 7.12. The label placement produced by Maplex that was configured to prevent overlapping of the roads and water bodies. Projection: spherical Mercator (EPSG:3857). Data source: Open-StreetMap project (2013). The red circles mark overlaps of labels with nearby point symbols.

visually compared the label placement quality for the overlap cases. We informally considered the amount of overprinting, the number and nature of overlaps and the legibility of names in relation to the surrounding geographic features. In our judgment our method accomplishes its task better in 30 cases compared to 4 cases where we deemed the Maplex result to be of higher quality. All other cases (33, ~50%) were either at the very same position or of equal quality. Generally the comparison indicates that the proposed method produces results in the tested context that are of at least equal quality compared to Maplex. It is worth noting that the resulting name placement in Figure 7.11 can be undoubtedly improved by using sliding label positions.

In the next set of experiments we investigated the effect of the visual contrast metric (Section 7.3.6) on the resulting labelling. For that purpose, we show the adaptability of our measure for labelling relief maps. For the test we chose the same region as depicted in Figure 7.2. This region is located on the border between three countries: Macedonia, Albania and Greece. As in the first experiment we used the parameterization which is defined in the first row of the set № 3. The result of this experiment is depicted in Figure 7.14. As we can see, some labels (marked with red frames) are difficult to search and to

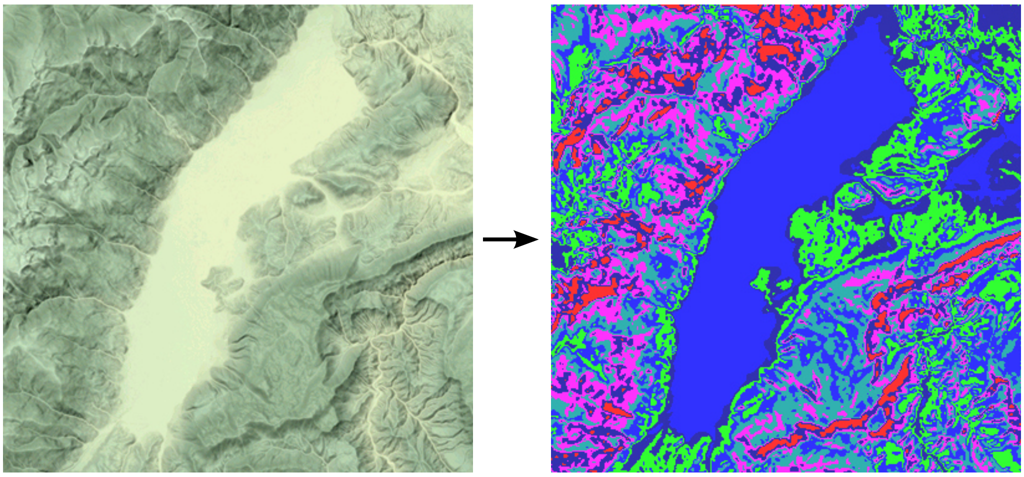


Figure 7.13. An example of shaded relief map segmentation using K-means algorithm with 8 clusters.

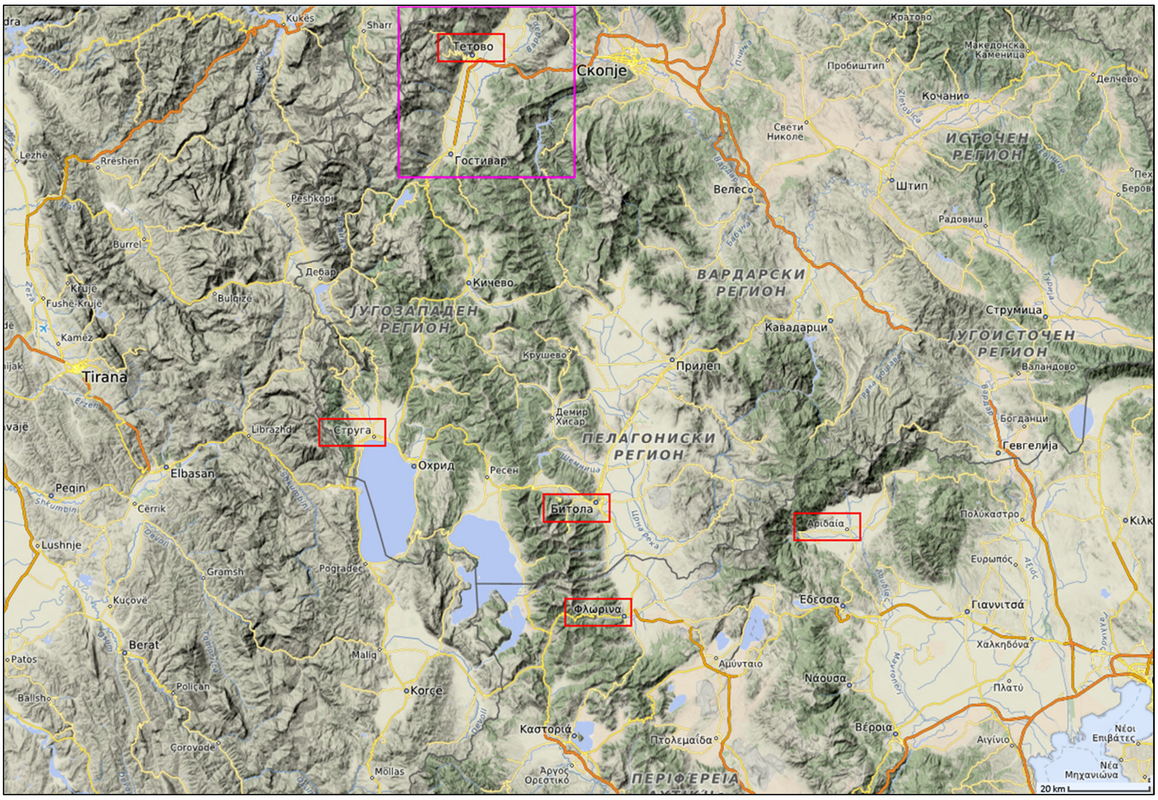


Figure 7.14. The labelling using the measure with the preference for background homogeneity and spatial distribution metrics. Projection: spherical Mercator (EPSG:3857). Data source: ASTER GDEM (2011) and OpenStreetMap project (2013). The purple rectangle shows the region depicted in Figure 7.13.

read. Many background elements in a map might lead to a darker colours accumulating in a cartographic product. One of the more popular background elements is hill-shading, which is bound to introduce more grey and black into the map, even if counteracted by sun-tones and lighter colours as done masterfully in the Swiss style (Imhof, 1982/2007). Automated analytic shading as it is available in many software packages has an an even stronger tendency to darken the overall map (e.g., Figure 7.13) (Jenny and Patterson, 2007). This can severely limit the legibility of the positioned text (Figure 7.2) even with letter casing (halo) (see Figure 7.14). In order to improve the labelling even for maps with a background with many dark elements, we tested the visual contrast component. We prepared the second background mask of the relief based on ASTER GDEM data (2011). As we have two masks, the measure (7.13) should be considered twice in equation (7.14). Namely, we split it into two constituent parts (see two rows of the parameter set № 3), where the first set was used for the background mask with roads and water bodies, and the second one for the mask with shaded relief respectively. The quality of type placement can be significantly enhanced by taking into account shaded relief and the visual contrast metric (Figure 7.15). In Figure 7.15 the previously marked labels are much more legible and have less overprinting of the surrounding shades in comparison to

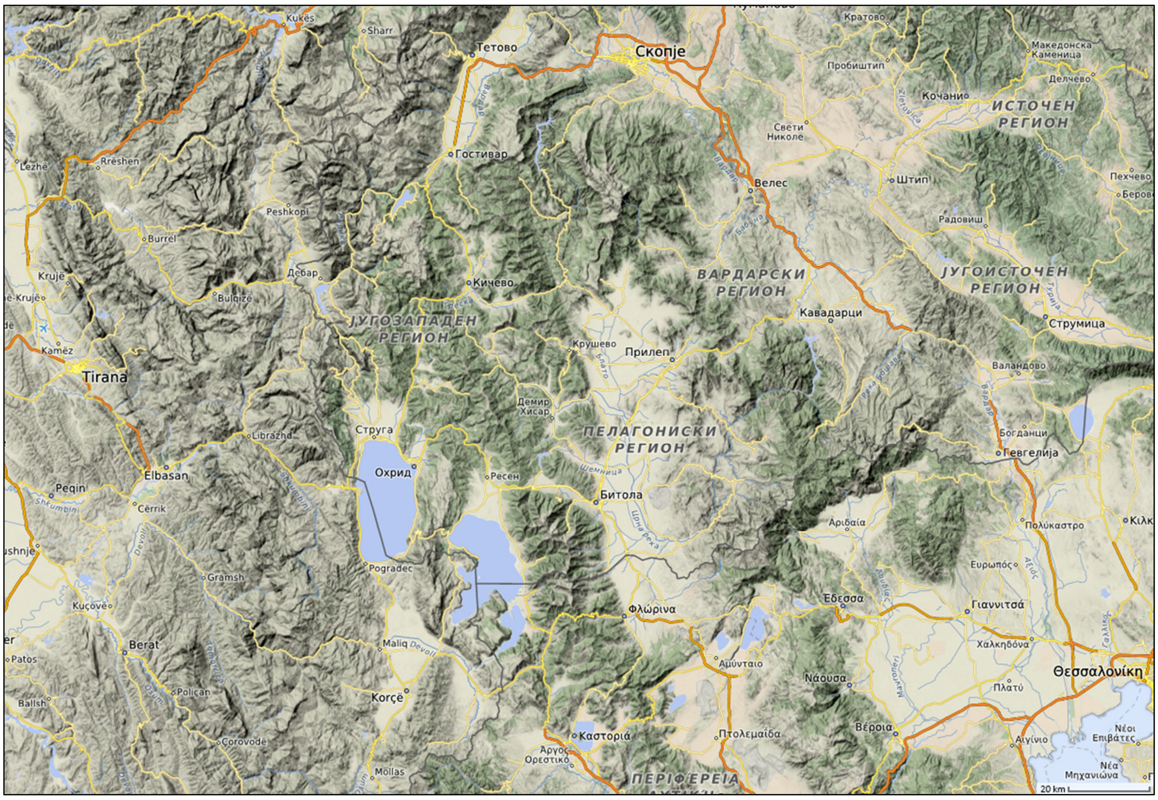


Figure 7.15. The labelling using all metrics. Projection: spherical Mercator (EPSG:3857). Data source: ASTER GDEM (2011) and OpenStreetMap project (2013).

the preceding placement in Figure 7.14. Also note, at the same time the labels have less interference with roads, rivers and lakes.

7.5 Summary and Conclusions

This paper introduced the measure for quantifying the quality of potential label positions based on map background information. The proposed measure is built upon some cartographic principles of type placement. As we have shown, a general labelling algorithm (Edmondson et al., 1996) equipped by our measure tends to locate names in map regions with lower graphic complexity. It also tries to minimize interference with background elements and to avoid visual clutter between graphical elements of similar colours. We performed a set of experiments on real-world datasets. The results of the reported experiments clearly indicate that applying our measure in automated map lettering can significantly improve name placement in respect of higher clarity and better legibility of the map, which indisputably make maps more effective. Our measure is well and sufficiently defined to be fully reproduced in and adopted to any other application. Moreover, the comparison of the presented method with a vector-based method implemented in Maplex indicates comparable quality of label placement in terms of feature overlap.

The overall computational complexity and memory consumption of our measure is low due to the raster representation instead of using vector geometries. It was shown in the experiments that our measure needs ~0.2 s to perform labels evaluation. This amount of time falls into the range of 1-2 seconds that is considered as reasonable for interactive cartography (Roth, 2013). This fact makes the proposed measure appropriate for use in interactive and dynamic labelling (Been et al., 2006; Mote, 2007) and especially attractive for labelling on mobile devices (Kovanen and Sarjakoski, 2013). However, a practical implementation of the measure in a real-time or mobile context would benefit from some modifications. These should include the parallel computation of scores for candidate positions on multiple independent CPUs and speeding up the image segmentation using a GPU. Note that the stage of image segmentation can potentially be performed on the server side in advance before the image is delivered to the client. In this case, the client also needs the table of corresponding colours. Moreover, the real-time context requires a rather fast algorithm for solving the combinatorial optimization problem. Another possible application for our method is automated labelling of 3D models (Götzelmann et al., 2005; Lehmann and Döllner, 2012).

The proposed quality measure was designed for and tested on point-feature label placement with axis-aligned labels. However, it is notable, that the presented approach can be modified to support evaluation of labelling for any placement scheme (e.g., points (Carstensen, 1987; Rylov and Reimer, 2014a), lines (Edmondson, 1996; Wolff et al.,

2001) or polygons (van Roessel, 1989; Barrault, 2001; Rylov and Reimer, 2014b)). The difference amounts to the need for the resampling of pixels that lie within an arbitrarily rotated rectangle describing the bounds of a letter.

The comparison and study of the influence of different image segmentation algorithms and their parameterization on resulting labelling were not conducted. However, these questions are still open and require further research.

Acknowledgements

The authors would like to warmly thank colleagues at our GIScience Research Group for productive discussions and their valuable feedback. We would like to express our particular appreciation to the anonymous reviewers whose constructive and helpful comments substantially improved the paper. Financial support as the scholarship from DAAD (Deutscher Akademischer Austauschdienst, German Academic Exchange Service) obtained by Maxim Rylov is gratefully acknowledged. Additional thanks go to Sarah Lohr for helping with the comparison of labelling results.

References

Adams, R., L. Bischof. 1994. Seeded region growing. Pattern Analysis and Machine Intelligence, IEEE Transactions on 16(6): 641-647. doi: 10.1109/34.295913

Ahn, J., and H. Freeman 1984. “A Program for Automatic Name Placement.” *Cartographica* 21(2-3): 101-109. doi: 10.3138/0646-Q262-6636-3681

ASTER Validation Team. 2011. ASTER Global Digital Elevation Model Version 2 – Summary of Validation Results. Published by the ASTER GDEM Validation Team: METI, NASA and USGS. August 2011, 27 pages. Available online at: http://www.jspacesystems.or.jp/ersdac/GDEM/ver2Validation/Summary_GDEM2_validation_report_final.pdf. http://reverb.echo.nasa.gov.

Barrault, M., and F. Lecordix. 1995. “An Automated System for Linear Feature Name Placement which Complies with Cartographic Quality Criteria.” In *Proc. AutoCarto 12:* 321-330. ACSM/ASPRS, Bethesda.

Battista, G. D., P. Eades, R. Tamassia, and I. G. Tollis. 1994. “Algorithms for Drawing Graphs: An Annotated Bibliography.” *Computational Geometry: Theory and Applications* 4(5): 235-282. doi: 10.1016/0925-7721(94)00014-X

Been, K., E. Daiches, and C. Yap. 2006. “Dynamic Map Labeling.” *IEEE Transactions on Visualization and Computer Graphics* 12(5): 773-780. doi: 10.1109/TVCG.2006.136

Brewer, C.A. 2005. Designing Better Maps: A Guide for GIS Users, ESRI Press, Redlands CA.

Brown, B. 1976. “Effect of Background Constraint on Visual Search Times.” *Ergonomics* 19(4):441-449. doi: 10.1080/00140137608931556

Carstensen, L. W. 1987. “A Comparison of Simple Mathematical Approaches to the Placement of Spot Symbols.” *Cartographica* 24(3): 46-63. doi: 10.3138/GH04-24H7-63T8-167V

Castner, H. W. and R. J. Eastman. 1985. “Eye-Movement Parameters and Perceived Map Complexity—II.” *American Cartographer* 12(1): 29-40. doi: 10.1559/152304085783914712

Christensen, J., J. Marks, and S.M. Shieber. 1995. “An Empirical Study of Algorithms for Point-Feature Label Placement.” *ACM Transactions on Graphics* 14(3): 203-232. doi: 10.1145/212332.212334

CIE. 1978. Recommendations on Uniform Colour Spaces, Colour Difference Equations, Psychometric Colour Terms. CIE Publication.

Cromley, R.G. 1985. “An LP Relaxation Procedure for Annotating Point Features Using Interactive Graphics.” In *Proceedings of the AutoCarto 7,* 127-132.

Defence Geographic and Imagery Intelligence Agency (UK) (1973) Series ONC, Sheet F-3, Albania, Bosnia, etc., 1:1000000, Edition 14-GSGS, Reprinted by DGIA, Ministry of Defence, United Kingdom, 2000. © UK MOD Crown Copyright, 2014. Scan provided by University of Texas, Perry-Castañeda Library Map Collection. http://www.lib.utexas.edu/maps/onc/txu-pclmaps-oclc-8322829_f_3.jpg (accessed on October 22, 2013)

Doerschler, J.S., and H. Freeman. 1992. “A Rule-Based System for Dense-Map Name Placement.” *Communications of the Association of Computing Machinery* 35(1): 68-79. doi: 10.1145/129617.129620

Edmondson, S., J. Christensen, J. Marks, and S. Shieber. 1996. “A General Cartographic Labeling Algorithm.” *Cartographica* 33(4): 13-23. doi: 10.3138/U3N2-6363-130N-H870

ESRI. (2009). ‘Maplex for ArcGIS. An ESRI White Paper’. Available at http://www.esri.com/library/whitepapers/pdfs/maplex-for-arcgis.pdf.

Fairbairn, D. 2006. “Measuring Map Complexity.” *Cartographic Journal* 43(3):224-238. doi: 10.1179/000870406X169883

Freeman, H. 1991. “Computer name placement.” In *Geographical Information Systems: Principles and Applications.* ed. D.J. Maguire, M.F. Goodchild, and D.W. Rhind. 445-456, London: Longman.

Freeman, H., and J. Ahn. 1987. “On the Problem of Placing Names in a Geographic Map.” *International Journal of Pattern Recognition and Artificial Intelligence* 1(1): 121-140. doi: 10.1142/S0218001487000096

García, R., J.P. de Castro, E. Verdú, M.J. Verdú, and L.M. Regueras. 2012. “Web Map Tile Services for Spatial Data Infrastructures: Management and Optimization.” In *Cartography - A Tool for Spatial Analysis*. ed. C. Bateira. 25-48. doi: 10.5772/46129

Gervautz, M., and W. Purgathofer. 1988. “A Simple Method for Color Quantization: Octree Quantization.” *New Trends in Computer Graphics*, ed. N. Magnenat-Thalmann and D. Thalmann. Springer Berlin Heidelberg. 219-231. doi: 10.1007/978-3-642-83492-9_20

Goodchild, M.F. 2007. “Citizens as Sensors: The World of Volunteered Geography.” *GeoJournal* 69: 211-221. doi: 10.1007/s10708-007-9111-y

Götzelmann, T., K. Ali, K. Hartmann, and T. Strothotte. 2005. “Form Follows Function: Aesthetic Interactive Labels.” In *Computational Aesthetics in Graphics, Visualization and Imaging*. 193-200. doi: 10.2312/COMPAESTH/COMPAESTH05/193-200

Haklay, M., and P. Weber. 2008. “OpenStreetMap – User Generated Street Map”, IEEE Pervasive Computing 7(4): 12-18. doi: 10.1109/MPRV.2008.80

Haralick, M. and Shapiro, L. G. 1985. “Image segmentation techniques.” *Computer Vision Graphics and Image Processing* 29: 100-132. doi: 10.1016/S0734-189X(85)90153-7

Harrie, L., H. Stigmar, T. Koivula, and L. Lehto. 2004. “An Algorithm for Icon Placement on a Real-Time Map.” In *Developments in spatial data handling*, ed. P. Fisher. 493-507. Berlin: Springer.

Harrie, L., and H. Stigmar. 2010. “Evaluation of Measures for Quantifying Map Information.” *ISPRS Journal of Photogrammetry and Remote Sensing* 65(3): 266-274. doi: 10.1016/j.isprsjprs.2009.05.004

Hirsch, S.A. 1982. “An Algorithm for Automatic Name Placement around Point Data.” *American Cartographer* 9(1): 5-17. doi: 10.1559/152304082783948367

Hong, F., Z. Zuxun, and D. Daosheng. 2005. “Quality evaluation model for map labeling.” *Geo-spatial Information Science* 8(1): 72-78. doi: 10.1007/BF02826996

Imhof, E. 1975. “Positioning names on maps.” *American Cartographer* 2(2): 128-144. doi: 10.1559/152304075784313304

Imhof, E. 1982. Cartographic Relief Presentation. Berlin: Walter de Gruyter.

Imhof, E. 2007. Cartographic Relief Presentation (reprint). Redmond: ESRI Press

Jain, A.K., M.N. Murty, and P.J. Flynn. 1999. “Data clustering: review.” *Journal ACM Computing Surveys* 31(3): 264-323. doi: 10.1145/331499.331504

Jenny, B., and L. Hurni. 2006. “Swiss-Style Colour Relief Shading Modulated by Elevation and by Exposure to Illumination.” *Cartographic Journal* 43(3): 198-207. doi: 10.1179/000870406X158164

Jenny, B., and T. Patterson. 2007. “Introducing Plan Oblique Relief.” *Cartographic Perspectives* 57:21-40.

Jones, C. B. 1989. “Cartographic Name Placement with Prolog.” *IEEE Computer Graphics and Applications* 9(5): 36-47. doi: 10.1109/38.35536

Jurio, A., H. Bustince, M. Pagola, P. Couto, and W. Pedrycz. 2013. “New measures of homogeneity for image processing: an application to fingerprint segmentation.” *Soft Computing*, Berlin: Springer. doi: 10.1007/s00500-013-1126-3

Kakoulis, K. G. and I. G. Tollis., 2003. “A Unified Approach to Automatic Label Placement.” *International Journal of Computational Geometry and Applications* 13(1): 23-59. doi: 10.1142/S0218195903001062

Kirkpatrick, S., Jr.C.D. Gelatt, and M.P. Vecchi. 1983. “Optimization by Simulated Annealing.” *Science* 220: 671-680. doi: 10.1126/science.220.4598.671

Kovanen, J., and L.T. Sarjakoski. 2013. „Sequential displacement and grouping of point symbols in a mobile context.” *Journal of Location Based Services* 7(2): 79-97. doi: 10.1080/17489725.2013.764024

Lee, S., M. Sips, H-P. Seidel. 2013. Perceptually Driven Visibility Optimization for Categorical Data Visualization. *IEEE Transactions on Visualization and Computer Graphics* 19(10): 1746-1757. doi: 10.1109/TVCG.2012.315

Lehmann, C., and J. Döllner. 2012. “Silhouette-Based Label Placement in Interactive 3D Maps.” *Cartography from Pole to Pole. Lecture Notes in Geoinformation and Cartography.* ed. M. Buchroithner, N. Prechtel, and D. Burghardt. 177-186. doi: 10.1007/978-3-642-32618-9_13

Leyk, S., and R. Boesch. 2010. “Colors of the past: color image segmentation in historical topographic maps based on homogeneity.” *GeoInformatica* 14(1): 1-21. doi: 10.1007/s10707-008-0074-z

Li, Z., and P. Huang. 2002. "Quantitative measures for spatial information of maps." *International Journal of Geographical Information Science* 16(7): 699-709. doi: 10.1080/13658810210149416

Lloyd, R. 1997. "Visual Search Processes Used in Map Reading." *Cartographica* 34(1): 11-32. doi: 10.3138/F342-1217-827R-0777

Mote, K. 2007. "Fast Point-Feature Label Placement for Dynamic Visualizations." *Information Visualization* 6(4): 249-260. doi: 10.1057/PALGRAVE.IVS.9500163

Noyes, L. 1980. "The Positioning of Type on Maps: The Effect of Surrounding Material on Word Recognition Time." *Human Factors: The Journal of the Human Factors and Ergonomics Society* 22(3): 353-360. doi: 10.1177/001872088002200309

Open Geospatial Consortium. 2006. OpenGIS Web Map Server Implementation Specification. Tech. Rep. OGC 06-042, ed. J. de la Beaujardiere, Available at http://portal.opengeospatial.org/files/?artifact_id=14416.

Open Geospatial Consortium. 2010. OpenGIS Web Map Tile Service Implementation Standard. Tech. Rep. OGC 07-057r7, ed. J. Masó, K. Pomakis, N. Julià. Available at http://portal.opengeospatial.org/files/?artifact_id=35326.

OpenStreetMap. 2013. A collaborative project for creating a free editable geo-database of the world. Data © OpenStreetMap contributors. Accessed on November 12, 2013. Available at http://openstreetmap.org.

Pal, S.K., and N.R. Pal. 1987. "Segmentation using contrast and homogeneity measures." *Pattern Recognition Letters* 5(4): 293-304. doi: 10.1016/0167-8655(87)90061-4

Pal, N.R., and S.K. Pal. 1993. "A review on image segmentation techniques." *Pattern Recognition* 29: 1277-1294. doi: 10.1016/0031-3203(93)90135-J

Phillips, R.J., L. Noyes, and R.J. Audley. 1977. "The Legibility of Type on Maps." *Ergonomics* 20(6): 671-682. doi: 10.1080/00140137708931679

Phillips, R.J., L. Noyes. 1982. "An Investigation of Visual Clutter in the Topographic Base of a Geological Map." *Cartographic Journal* 19(2): 122-132. doi: 10.1179/caj.1982.19.2.122

Purves, D., and R.B. Lotto. 2010. "Why We See What We Do Redux: A Wholly Empirical Theory of Vision." 2nd ed. Sinauer Associates, Inc.

Ramm, F., J. Topf, and S. Chilton. 2010. *OpenStreetMap: Using and Enhancing the Free Map of the World*. Cambridge: UIT Cambridge.

Reimer, A. 2011. “Squaring the Circle? Bivariate Colour maps and Jacques Bertin's concept of 'Disassociation' (Digital Proceedings),” In Proceedings of the 25th International Cartographic Conference, Paris, France.

Robinson, A.H., J.L. Morrison, P.C. Muehrcke, A.J. Kimerling, and S.C. Guptil 1995. *Elements of Cartography*, 6th ed. New York: Wiley.

Roth, R.E. 2013. “Interactive maps: What we know and what we need to know.” *Journal of Spatial Information Science* 6: 59-115. doi: 10.5311/JOSIS.2013.6.105

Rylov, M.A., and Reimer, A.W. 2014a. “A Comprehensive Multi-Criteria Model for High Cartographic Quality Point-Feature Label Placement.” *Cartographica* 49(1): 52-68. doi: 10.3138/carto.49.1.2137.

Rylov, M.A., and Reimer, A.W. 2014b “A Practical Algorithm for the External Annotation of Area Features.” *Cartographic Journal.* doi: 10.1179/1743277414Y.0000000091

Schrijver, S. 2003. Combinatorial Optimization: Polyhedra and Efficiency. Berlin: Springer.

Shannon, C.E., and W. Weaver. 1964. “The Mathematical Theory of Communication.” Champaign: University of Illinois Press.

Shapiro, L.G., and G.C. Stockman. 2001. “Image segmentation”. In *Computer Vision.* 279-325, New Jersey: Prentice-Hall.

Stadler, G., T. Steiner, and J. Beiglbock. 2006. “A Practical Map Labeling Algorithm Utilizing Morphological Image Processing and Force-Directed Methods.” *Cartography and Geographic Information Science* 33(3): 207-215. doi: 10.1559/152304006779077327

Strijk, T.W., and M.J. van Kreveld. 2002. “Practical Extensions of Point Labeling in the Slider Model”, *GeoInformatica* 6(2): 181-197. doi: 10.1023/A:1015202410664

Sukhov, V.I. 1967. “Information capacity of a map entropy.” Geodesy and Aerophotography 10: 212–215.

Sukhov, V.I. 1970. “Application of information theory in generalization of map contents.” International Yearbook of Cartography, X: 41–47.

van Dijk, S., M.J. van Kreveld, T.W. Strijk, and A. Wolff. 2002. “Towards an Evaluation of Quality for Names Placement Methods.” *International Journal of Geographical Information Science* 16(7): 641-661. doi: 10.1080/13658810210138742

van Kreveld, M.J., T.W. Strijk, and A. Wolff. 1999. “Point Labeling with Sliding Labels.” *Computational Geometry: Theory and Applications* 13: 21-47. doi: 10.1016/S0925-7721(99)00005-X van Roessel, W. 1989. “An Algorithm for Locating Candidate Labeling Boxes Within a Polygon.” *American Cartographer* 16(3): 201-209. doi: 10.1559/152304089783814034

Verner, O.V., R.L. Wainwright, and D. Schoenefeld. 1997. “Placing Text Labels on Maps and Diagrams using Genetic Algorithms with Masking.” *INFORMS Journal on Computing* 9(3): 266-275. doi: 10.1287/ijoc.9.3.266

Williams, L.G. 1967. “The effects of target specification on objects fixated during visual search.” *Acta Psychologica* 27: 355-360. doi: 10.3758/BF03207398

Wojtach, W.T. 2009. “Reconsidering Perceptual Content.” *Philosophy of Science* 76(1): 22-43.

Wolff, A., L. Knipping, M.J. van Kreveld, T.W. Strijk, and P.K. Agarwal. 2001. “A Simple and Efficient Algorithm for High-Quality Line Labeling.” *Technical Report UU-CS-2001-44*, Department of Computer Science, Utrecht University.

Wolff, A., and T. Strijk, 2009. “A map labeling bibliography.” Available at http://i11www.iti.uni-karlsruhe.de/~awolff/map-labeling/bibliography/.

Wood, C.H. 1994. “Effects of Brightness Difference on Specific Map-Analysis Tasks: An Empirical Analysis.” *Cartography and Geographic Information Science* 21(1): 15-30. doi: 10.1559/152304094782564019

Yamamoto, M., G. Camara, and L.A.N. Lorena. 2002. “Tabu Search Heuristic for Point-Feature Cartographic Label Placement.” *Geoinformatica* 6(1): 77-90. doi: 10.1023/A:1013720231747

Yoeli, P. 1972. “The Logic of Automated Map Lettering.” *Cartographic Journal* 9(2): 99-108. doi: 10.1179/000870472787352505

Zhang, Q. and L. Harrie. 2006a. “Real-Time Map Labelling for Mobile Applications.” *Computers, Environment and Urban Systems* 30(6): 773-783. doi: 10.1016/j.compenvurbsys.2006.02.004

Zhang, Q. and L. Harrie. 2006b. “Placing Text and Icon Labels Simultaneously: A Real-Time Method.” *Cartography and Geographic Information Science* 33(1): 53-64. doi: 10.1559/152304006777323127

Zoraster, S. 1986. “Integer Programming Applied to the Map Label Placement Problem.” *Cartographica* 23(3): 16-27. doi: 10.3138/9258-63QL-3988-110H

8 Pairwise Line Labelling of Geographic Boundaries: An Efficient and Practical Algorithm

Authors Maxim A. Rylov and Andreas W. Reimer

Journal *Cartographic Perspectives*

History Revised: 16.08.2014
Submitted: 10.03.2014

Reference Rylov, M.A., and A.W. Reimer. 2014. “A Pairwise Line Labelling of Geographic Boundaries: An Efficient and Practical Algorithm.” *Cartographic Perspectives*, (submitted)

8 Pairwise Line Labelling of Geographic Boundaries: An Efficient and Practical Algorithm

Abstract

We present an algorithm that labels linear features with two matched toponyms describing the left and the right side of a line respectively. Such a pairwise line labelling is a common technique used in manually produced maps. The lines differentiate administrative divisions or other geographic subdivisions. Our approach solves two basic tasks of the automated map labelling problem, namely candidate-position generation and position evaluation for a given scale. The quality of the name placement is evaluated by comparison to a set of established cartographic principles and guidelines for linear features. We give some results of our experiments based on real datasets. The implementation of our algorithm shows that it is simple and robust, and the resulting sample maps demonstrate its practical efficiency.

Keywords: automated label placement, automated cartography, quality evaluation, computational geometry, GIS mapping

8.1 Introduction

Over the past few decades, there have been many attempts to automate label placement task in the field of cartography. Label placement algorithms have matured from being only able to solve the simplest problems (Yoeli, 1972; Basoglu, 1982; Hirsch, 1982) towards complex and sophisticated tools (e.g., ESRI's Maplex Label Engine, Maptext's Label-EZ, etc.) that are used in map production; see the excellent bibliography of papers on this topic maintained by Wolff and Strijk (2009). The main goal of labelling algorithms is to reduce manual work of a human cartographer by relieving him from the two basic tasks, namely:

- The editing of the map, i.e. the process of the name contents of maps.
- The actual positioning of the names on maps using predefined typefaces.

As a consequence, automated type placement reduces map production time and cost. Although commercial labelling packages have been available for some time, there is still a great need to manually resolve conflicts and introduce yet non-automated labelling techniques in order to achieve a professional level of functionality and legibility of the final map. In addition, commercial labelling packages are often difficult to parameterize to match production standards (Revell et al., 2011; Regnauld et al., 2013).

In cartography, all map objects to be labeled could be divided into three categories (Imhof, 1962/1975; Wood, 2000; Brewer, 2005): punctiform (e.g., settlements, mountain peaks), linear (e.g., roads, rivers, boundaries) and areal (e.g., countries, lakes, islands) designations. Each type of designation has its own requirements and involves its own challenges. Compelling attempts to automate map lettering were made by Yoeli (1972), Christensen et al. (1995), van Kreveld et al. (1999) for point features, by Barrault and Lecordix (1995), Edmondson et al. (1996), Chirié (2000), Wolff et al. (2001) for lines and by van Roessel (1989), Barrault (2001), Rylov and Reimer (2014b) for areas. In this article we propose a method for labelling a special type of linear features. We are interested in the pairwise labelling of linear features that demarcate area boundaries. There are several use cases, where the boundary itself needs to be labeled twice, differently on each side of the linear feature, e. g. international borders, municipal divisions, grid-zones or military zonings where different rules of engagement apply and so forth. In manual cartography two different design techniques are used to letter a boundary line in pairwise manner. More exactly, the boundary can be labeled either with a text placed along a straight line (Figure 8.1) or the names could be curved following the direction of the polyline to be annotated (Figure 8.2). Curved lettering often can be the preferred choice, aesthetics-wise. This paper presents an algorithm that is able to position labels in a way which is visually similar to the approach used in Figures 8.2 and 8.3b, namely when the label is not curved.

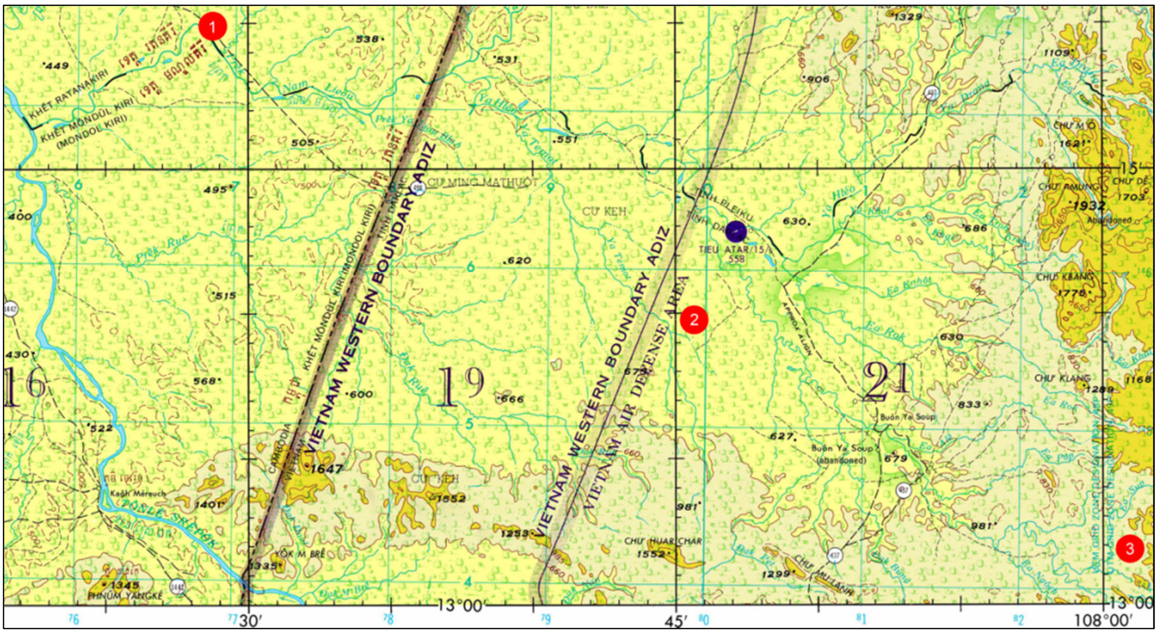


Figure 8.1. Example for manual usage of pairwise line labelling: provincial boundaries (bilingual) [1], zones for changing rules of engagement [2] and UTM grid zone changes [3], SOURCE: Defense Mapping Agency (1973).

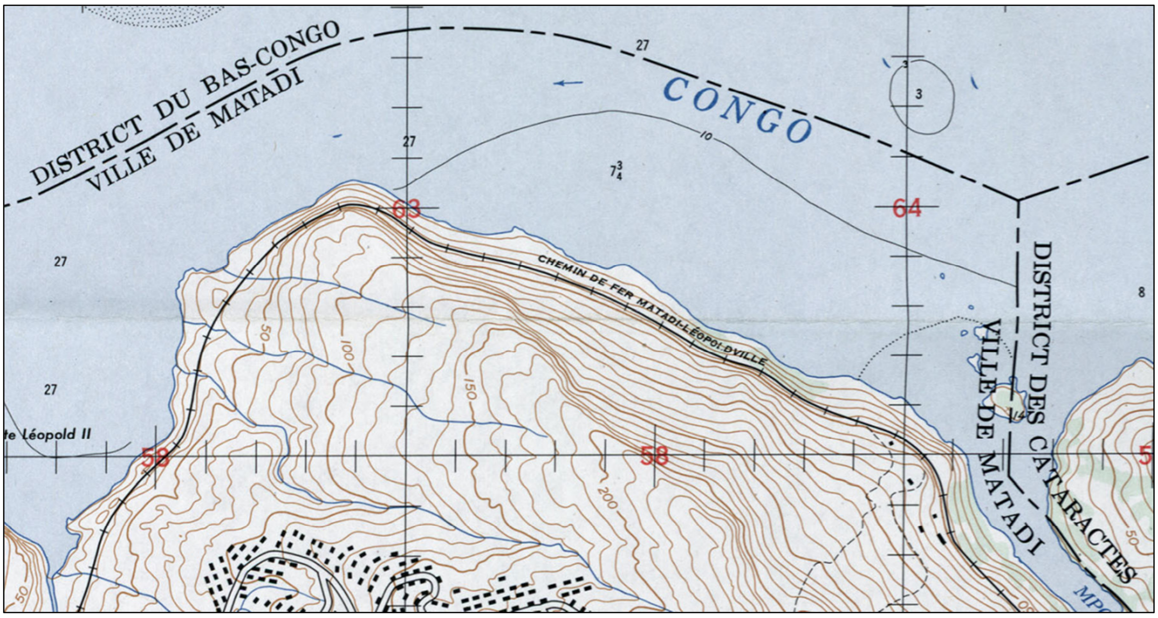
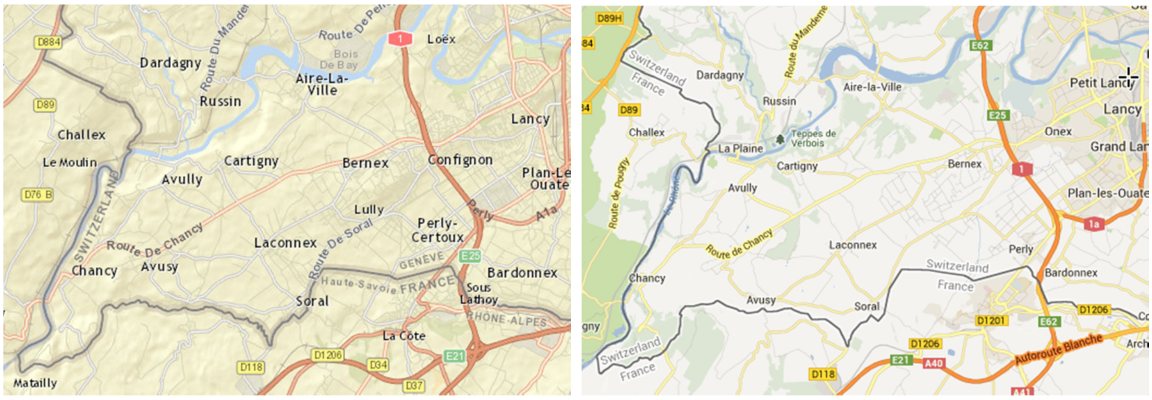


Figure 8.2. An example of applying curved labelling to annotate administrative boundaries. Source: National Imagery and Mapping Agency (1962).

The proposed algorithm can be used on large-scale maps for labelling of areas when the scale becomes too large to label these features as areas, by placing the label inside the areal object. With pairwise line labelling, regions that lie on opposite sides of a boundary line can be identified without difficulty. The main visual advantage is that a map reader is informed about the exact nature of the line, not only its general type. This helps to easily distinguish boundaries from other linear objects and amplifies the precise graphic relation between the toponyms and the relevant map features. Another strong feature of such labelling is the consideration of two names as a unit or a single label. It means that the resulting map is free from partial designations, i.e. a label either on the left or on the right side of the object (Figure 8.3a).

To the best of our knowledge, there are no preceding published works regarding automated pairwise line labelling. However, it is worth noting that some existing commercial label engines have the ability to label administrative boundaries. For instance, the Maplex Label Engine produces labelling of administrative units for each side of a linear feature independently (Figure 8.3a). Basically, this kind of labelling can be performed using any line labelling algorithm (e.g., Barrault and Lecordix, 1995; Edmondson et al., 1996; Chirié, 2000; Wolff et al., 2001). Note that such label placement is not widely used in traditional cartography and is in violation of cartographic principles about labelling from the literature and extant topographic maps. For example, this approach often creates ambiguities between the labels which annotate the boundaries of different subdivision levels (Figure 8.3a, “GENEVE & FRANCE”). The next example in



(a) (b)

Figure 8.3. An example of different lettering of administrative boundaries. (a) Non-pairwise line labelling of boundaries with mixed hierarchies. Map data © Esri. (b) Pairwise line labelling, closed source and unknown parameterization. Map data © 2014 Google.

Figure 8.3b illustrates map labelling on Google Maps, where the labels of national borders are coupled and positioned in the regions with less curvature of a line and where the text is less sloped. The two presented approaches follow different cartographic precepts, if at all. We interpret both approaches as arising from technical and theoretical limitations. The description and implementation of both mentioned algorithms is, of course, not known and closed source. We have found no free/open source label engine or research publication aimed at pairwise label placement.

We start describing our method in the following section with a formalization of the criteria representing the cartographic guidelines for pairwise line labelling. Next, we introduce a general form of our scoring function (van Dijk et al., 2002). Then, we continue with a description of the first part of the algorithm that consists of a method for generating of a set of potential label positions for each linear feature. Subsequently, we describe each part of the quality function in detail. The proposed quality measures take into account:

- the curvature of the polyline;
- the offset of label from the polyline;
- the orientation of the lettering;
- an even distribution of the labels along the polyline.

In general, a quality evaluation, or an objective function, can be employed by any combinatorial optimization algorithm (Christensen et al. 1995, Rabello et al. 2014) for finding a feasible near-optimal solution of the automated label placement problem. Note that the characteristics, like the position and the quality assessments, of the output label candidates can be used as input to a general map labelling algorithm (Edmondson et al., 1996; Kakoulis and Tollis, 1998) that is basically much more comprehensive and sophisticated. For example, these algorithms should consider figure-ground relationship (Rylov and Reimer, 2014c) or resolve any ambiguities between neighboring labels (Rylov and Reimer, 2014a). In the section with the results of our experiments we illustrate some significant map samples based on real-world datasets. These sample maps are labeled using our implementation of the proposed method. Finally, we conclude with a brief analysis of the present work and give some insights for future research.

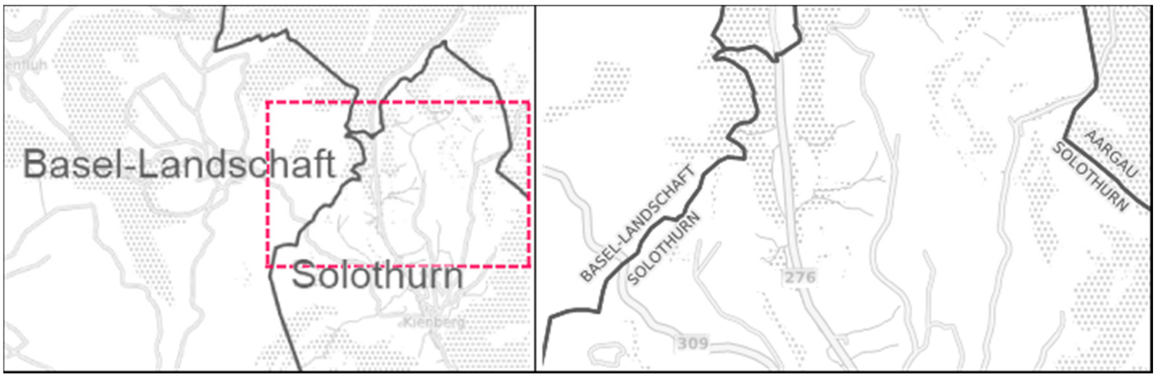
8.2 Method

The basic idea of the presented algorithm is depicted in Figure 8.4b and will be discussed in detail in the following sections.

Succinctly, the necessary input of our algorithm is a polyline that describes a boundary and two toponyms which define adjacent areal features. The output is a set of coupled labels that represent either side of the polyline to be annotated.

8.2.1 Approach Methodology

Automated text placement, or lettering, is one of the most difficult and complicated problems to be solved in automated cartography and Geographic Information Systems. When it comes to solving a complex problem, usually the problem is decomposed into smaller and simpler sub-problems. In our approach we use the same technique. Thus, the map labelling problem in general can be divided into three substantially independent subtasks (Edmondson et al., 1996). They are:



(a) (b)

Figure 8.4. Exemplary labelling of administrative regions in Switzerland. (a) Placement of the names inside the areas; (b) Positioning of the names along the boundary line.

- Candidate-position generation: A method that generates a set of label candidates for each map feature, using its spatial characteristics and taking into account its type (e.g., point, line or area). The generated potential label positions are normally considered as the search space for the position selection procedure.
- Position evaluation: A process of computing a score for each label candidate. This score is calculated using a quality function, which measures how well a label is positioned with respect to the object it tags as well as to other labels and features (van Dijk et al., 2002) on the map. In general, the quality function should take into account and reflect formal cartographic precepts applied to a certain type of designation (Rylov and Reimer, 2014a for point features).
- Position selection: A process of choosing only one label position from each set of candidates so that the total label quality measured with the quality evaluation function is globally maximized. The main goal of such processes consists in finding the maximum possible value of the quality function and thereupon to achieve a superior level of cartographic quality of the resulting map (Christensen et al., 1995). Note that the selection is an NP-hard problem in general (Formann and Wagner, 1991; Marks and Shieber, 1991).

Our algorithm deals with the two first subtasks of automated map lettering for the case of pairwise line labelling. Once these subtasks are solved, the position selection procedure can be applied. The position selection is canonically treated as a general optimization problem via strategies such as exhaustive search methods (Hirsch, 1982; Yoeli, 1972), simulated annealing (Edmondson et al., 1996), genetic algorithms (Verner et al., 1997), gradient based optimization (Christensen et al., 1995) or tabu search (Yamamoto et al., 2002). We describe the solution of these subtasks in the next three subsections. But first, we define and enumerate requirements for pairwise line labelling according to corresponding cartographic guidelines.

8.2.2 Linear Feature Labelling Requirements

We have selected and operationalized the relevant rules for pairwise line labelling from the extant cartographic literature on positioning names on maps (Imhof, 1962/1975; Yoeli, 1972; Wood, 2000; Brewer, 2005). The list of design rules adapted to our problem is as follows:

G1. A label must be placed along the linear feature it tags.

G2. A label should conform to the curvature of the polyline.

G3. Avoid complicated and extreme curvatures of the polyline. Straight or almost straight parts of the polyline should be preferred.

G4. A label must be placed close to the polyline, but not too close.

G5. The name must not be spread out, but may be repeated at specified intervals along the linear feature.
G6. Avoid placing names near end points of the polyline.
G7. Horizontally aligned labels are preferred to vertical ones.
G8. The two parts of a label should be centered relatively to each other.
G9. The name should not cross the linear feature.

The term "label" in the list actually means a "pair of labels", in other words, one label to annotate the left side of the polyline and another label for the right side respectively.

These guidelines are used as the criteria for candidate-position generation as well as for the position evaluation task in the following up subsections. Note that G2 is a general guideline which refers to different methods of lettering depicted in Figures 8.2 and 8.3b. Our approach deals with a technique when the text is straight, i.e. not curved.

8.2.3 Scoring Labelling Quality

When a potential position of a label is computed, it is numerically scored using a quality evaluation function. A quality function (van Dijk et al., 2002) achieves two main goals: to evaluate generated label positions regarding the cartographic precepts and to compare various labelling algorithms. Normally, a quality function is defined as a weighted sum of single metrics (Zhang and Harrie, 2006) and has the general form:

$$\begin{gathered} Q(L) = \sum_{l \in L} \big(w_1 \cdot f_{\text{priority}}(l) + w_2 \cdot f_{\text{aesthetics}}(l) + \\ + w_3 \cdot f_{\text{association}}(l) + w_4 \cdot f_{\text{label-visibility}}(l) + w_5 \cdot f_{\text{feat-visibility}}(l)\big) \end{gathered} \tag{8.1}$$

where $L = (l_1, \dots, l_n)$ is a set of n labels on the map, w_i $(i = 1, \dots, 5)$ are the weights and $f_*(l)$ are the quality metrics that measure how good the demands of cartographic guidelines are met in the positioning of a label l. Basically, the return value in (8.1) is usually normalized to the range [0,1]. For a detailed description and the meaning of each partial metric $f_*(l)$ we refer to the work by van Dijk et al. (2002). In addition, a review paper by Kern and Brewer (2008) contains a comparison table, which shows how the four criteria $f_{\text{aesthetics}}(l)$, $f_{\text{association}}(l)$, $f_{\text{label-visibility}}(l)$ and $f_{\text{feat-visibility}}(l)$ have been used in various proposed techniques and algorithms presented in the literature.

In equation (8.1) the measure $f_{\text{aesthetics}}(l)$ evaluates the quality of the position and the shape of a label with respect to the geometry of the feature it annotates, $f_{\text{association}}(l)$ defines how clear the association between a feature and its label is. In our approach, we construct a quality function (measure) called $F(l)$, which tutes $f_{\text{aesthetics}}(l)$ and partially $f_{\text{association}}(l)$. With a new function we want numerically score potential label positions, which are the output of the method presented in the next section. The components of the measure are the metrics that are designed to meet the requirements of some of the cartographic rules that we specified in the previous section. Let us define $F(l)$ for scoring l by analogy with (8.1) as:

$$\begin{aligned} F(l) = m_1 \cdot g_{\mathrm{PosDev}}(l) + m_2 \cdot g_{\mathrm{BaseOffset}}(l) + \\ + m_3 \cdot g_{\mathrm{GoodnessOfFit}}(l) + m_4 \cdot g_{\mathrm{HorizAlign}}(l) \end{aligned} \tag{8.2}$$

where m_1, m_2, m_3, m_4 are the weight factors, and $g_*(l)$ functions are:

- $g_{\mathrm{PosDev}}(l)$ measures the deviation of a label position from an even distribution of labels along the polyline.
- $g_{\mathrm{BaseOffset}}(l)$ evaluates how far is a label, which is settled on the baselines, from the centerline (Figure 8.5).
- $g_{\mathrm{GoodnessOfFit}}(l)$ represents a measure for quantifying how good the centerline approximates the polyline in a given region.
- $g_{\mathrm{HorizAlign}}(l)$ evaluates the deviation of the orientation of the label from a horizontal alignment which is parallel to the neat line of the map.

Note that the weights m_1, m_2, m_3, m_4 should sum up to 1.0 and the return values of each corresponding metric should fall into the range [0,1]. We design our metrics to yield higher values for label positions that are closer to the ideal position. To examine other research attempts, which deal with developing of quality measures to quantify label positions for linear features, we refer to the works by Barrault and Lecordix (1995), Edmondson et al. (1996) and Chirié (2000).

8.2.4 Candidate-Position Generation

In this section we introduce an algorithm that produces candidate positions along the input polyline (Figure 8.5). The algorithm produces a set of imaginary line segments (see centerline in Figure 8.5) which locally approximate the original polyline in a certain region. The width of the region (grey area) equals the maximum width of the two names. In addition to the centerline, an offset of the label (see d in Figure 8.5) from the polyline is computed. The centerline and the offset define two baselines. The baseline is the line upon (or under) which the characters of the name are drawn. Note that the candidate-position generation algorithm complies with the guidelines G1, G2, G4, G5 and G6.

We define some useful terms and measures before giving a detailed description of the algorithm. The input of our algorithm consists of a polyline $P = (p_1, \dots, p_n)$ specified by a sequence of points $p_i = (x_i, y_i)$, where $i = 1, \dots, n$ (Figure 8.6); and two names n_{l} and n_{r} that describe the left and the right side of the polyline P. We denote the total length of P by L. Let w_{l} and w_{r} be the widths (in map units) of n_{l} and n_{r} respectively. In order to satisfy requirement G5, we introduce a parameter S that defines the distance between

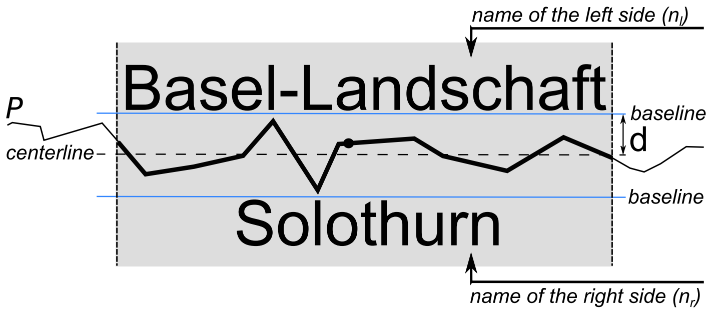


Figure 8.5. Sketch for the nomenclature used in describing the candidate-position generation algorithm.

names repeated along the polyline (Figure 8.6). We define the width of a label as $w_{\max} = \max(w_{\mathrm{l}}, w_{\mathrm{r}})$. The algorithm is composed of four phases that are detailed below.

8.2.4.1 Phase I

In the first phase, we generate a set of candidate locations along the polyline P (G5). We denote a point that represents the anchor point of a candidate position by q_j, where $j = 1, \ldots, m$ and $m = \lceil (L - S)/S \rceil$ is the number of such points. The point q_j lies on P and its distance from the starting point of P is defined by $S'(j) = (1/2 + j) \cdot S$.

Let q_j, $j = 1, \ldots, m$, be the points at which we are going to construct the centerline for placing a label. We consider the points q_j as preliminary locations as they are different from the resultant ones. The explanation of the difference between them is provided below. Next, in order to increase the size of the search space, we move each point q_j along the polyline in both directions until the distance from q_j along P reaches a certain value, the maximum position deviation $D_{\max}$. This approach gives us a set of positions (see grey areas in Figure 8.6) $\bar{q}_{jk}$, where $k \in [-N\ , N]$ with the centre at q_j. We denote this set by V_j. N is the half of the number of preliminary locations in V_j and defined as $N = \lceil D_{\max}\ / D_{\mathrm{step}} \rceil$, where D_{step} is the distance between two points $\bar{q}_{jk}$ and $\bar{q}_{jk+1}$. The total number of preliminary locations is calculated as $N_{\mathrm{total}} = 2 \cdot N \cdot m = 2 \cdot N \cdot \lceil \frac{L-S}{S} \rceil$. Assume that each point $\bar{q}_{jk}$ specifies a rough position for a label placement. Therefore, the maximum number of labels for one linear feature is equal to m, as only one label from each set V_j can be chosen. The adjustment of $D_{\max}$ and D_{step} should be done by the user, which controls the size of the search space. The parameters $S, D_{\max}$ and D_{step} are measured in map units.

As the allowed position deviation $D_{\max}$ increases, the distribution of labels along P becomes less regular. It can be seen in Figure 8.6 that the method for candidate-position generation also complies with G6 (avoidance of end points) automatically.

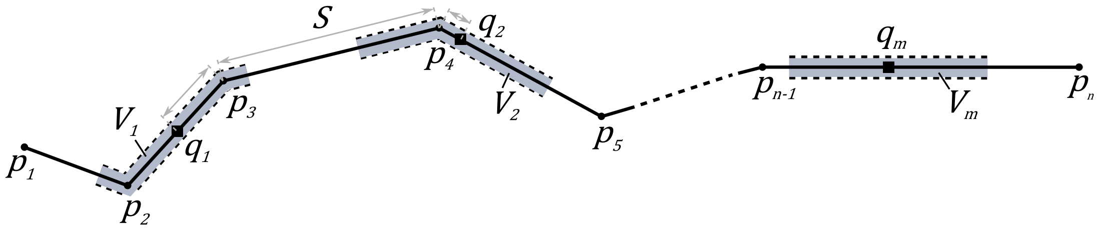


Figure 8.6. The input polyline P (solid black line) with nodes p_i, q_j are the centers of sets of potential locations V_j, S – the interval between q_j and q_{j+1}.

8.2.4.2 Phase II

In this phase, we try to find a centerline which approximates a part of P centered at $\bar{q}_{jk}$. Each part of this kind consists of points whose distance from $\bar{q}_{jk}$ along P is at most at $w_{\max}/2$. Such a centerline, or the best-fitting straight line denoted as R_{p}, can be found by employing the method of least squares (Chatterjee and Hadi, 2006). This method requires a set of points as the input. An approach of finding this set of points on P and consequently line R_{p} is described in the following steps.

(1) Let C be a circle with the centre at $\bar{q}_{jk}$ (Figure 8.7) and a radius equal to $r_{\mathrm{c}} = K \cdot w_{\max}$, where K is a control parameter that has values in the range [0.5,1]. As the actual shape of P is unknown, the circle radius is grown by increasing K until a satisfactory solution is found, i.e. step 3 has been passed and a centerline was found. Next, we want to find points of intersection between P and C. Due to the possible sinuosity of P, there could be many such points. Therefore, we consider only those two points of intersection whose distance from $\bar{q}_{jk}$ along P is the shortest. These two points we denote as t_1 and t_2. Note that there are two special cases when it is not possible to find these points:

- P fully lies inside the circle C.
- $\bar{q}_{jk}$ is too close to the one of the ends of P.

The distance between t_1 and t_2 should be large enough to accommodate the label. Therefore, we check if the distance is less than $w_{\max}$ before moving on to the next step. A refinement step can also be applied by trying several circles with different radii, as the curvature of a line can vary greatly from a straight line to a very bent curve.

(2) Construct the best-fitting straight line R_{p} from a set of points. This set consists of all vertices of P that lie inside the circle C with the centre at $\bar{q}_{jk}$ and radius r_{c}. In Figure 8.7 these points are: t_1, p_{i-1}, p_i and t_2. R_{p} is a preliminary line.

(3) We check whether P reverts too far back on itself for label placement, i.e. whether it represents a bulge in the segment under consideration. For this, we construct

the perpendicular to R_p through the point $\bar{q}_{jk}$ and check whether the points t_1 and t_2 are on the same side of the perpendicular. If the points t_1 and t_2 happen to be on the same side of the perpendicular, we consider R_p to be invalid. In this case we skip $\bar{q}_{jk}$ and move to the next point $\bar{q}_{jk+1}$ and repeat steps 1–3.

8.2.4.3 Phase III

Every time the circle C is grown beyond $K = 0.5$, the Euclidean distance between t_1 and t_2 can be greater than w_max. In this case we assume that R_p is not optimal and consider it as a first approximation. Therefore, we describe the procedure that refines the result of Phase II.

(1) Construct a perpendicular from the point $\bar{q}_{jk}$ to R_p. Find a point $q_{R\text{p}}$ that is the intersection of the perpendicular and R_p.

(2) Find two perpendiculars to R_p that are equidistant from the point $q_{R\text{p}}$. The distance between $q_{R\text{p}}$ and each of them is $0.5 \cdot w_\text{max}$.

(3) Find the points of intersection between P and the perpendiculars from step 2. Denote these points as s_1 and s_2 respectively (Figure 8.7).

(4) Find the best-fitting straight line R from a new set of points s_1, p_{i-1}, p_i, s_2 (Figure 8.8).

(5) Construct a perpendicular from the point $\bar{q}_{jk}$ to R. The point of intersection we denote as q_R. This point defines the centre of a label that will be placed along the centerline R.

Note that Phase III should be omitted if $K = 0.5$.

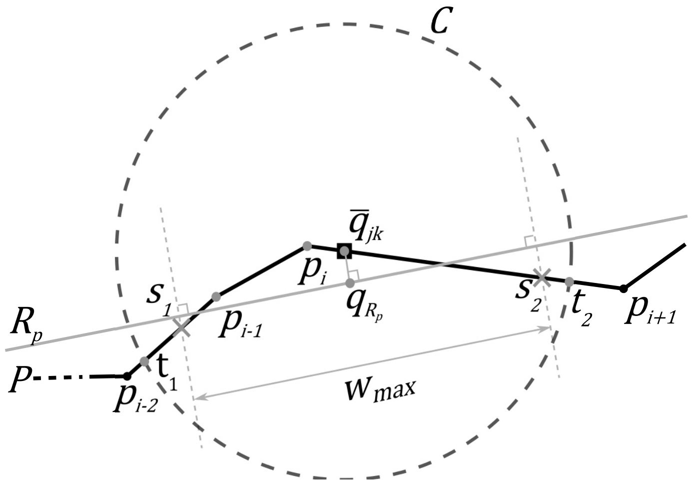


Figure 8.7. Best-fitting straight line R_p to a set of points of P with its centre in $\bar{q}_{jk}$.

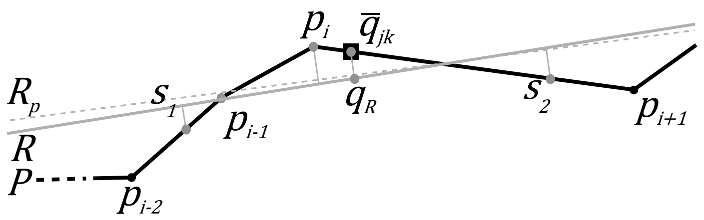


Figure 8.8. Refinement of the preliminary centerline R_p.

8.2.4.4 Phase IV

The final phase computes the offsets for the baselines which the labels will be placed upon (or under, see Figure 8.5). This phase is presented in three steps.

1. Compute the Euclidean distance between R and each point in the set of points that we have employed for constructing R. Put the values of the distances into two separate lists. The first list contains the points that lie on the left side from R and the second list for the points of the right side, respectively.
2. Compute the maximum value of all entries in each list. These values denoted as h_l and h_r are the offsets of the baselines BL_l and BL_r from the centerline R (Figure 8.9). Each offset defines the Euclidean distance of the respective baseline to R.
3. We increase each offset from the centerline by adding a typeface-dependent value to each offset. This approach helps to avoid overlapping of the polyline with the descenders or the ascenders of the characters of the label, e.g., $h_\mathrm{l} = 0$ and $h_\mathrm{r} = 0$.

To comply with the condition of G8, the names should be centered in regard to q_R.

8.2.4.5 Output of Phases I-IV

After applying four phases for each point $\bar{q}_{jk}$, a candidate label position can be defined with the following properties:

- The centre point q_R.
- The angle between the centerline and horizontal axis, denoted as α.
- Two offsets h_l and h_r from the centerline that represent the baselines for placing the characters.
- The coefficient of determination (explained below), denoted as γ, that was computed from the same set of points that we used for computing the centerline R. The value of this coefficient will be used later as one of the input parameters for the quality measure.

These outputs are used for the evaluation of candidate positions.

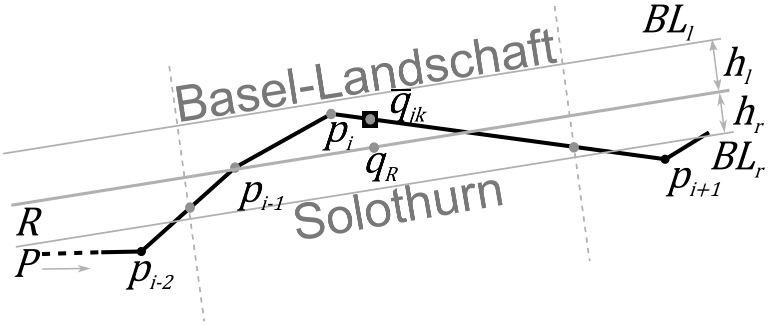


Figure 8.9. Computation of baseline offsets h_l, h_r from the centerline R.

We described the position generation procedure in four phases. As a consequence, we can conclude that the output of the presented method meets the requirements of six cartographic guidelines listed above (see Section 8.2.2), namely the rules G1, G2 in Phases II-III, partially G3 in Phase III (see Step 4), G4 in Phase IV. Guidelines G5 and G6 are fulfilled automatically in the approach for computing of q_j (Phase I). In the following sections we also use G4 and G5 for scoring label candidates.

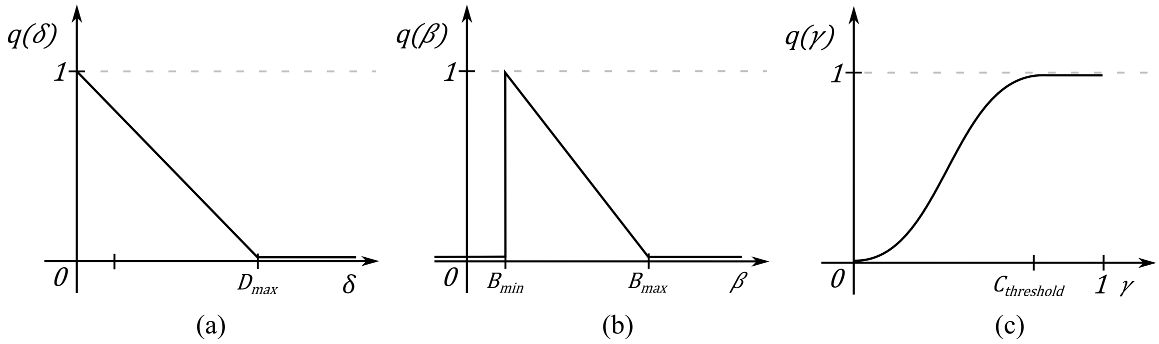
8.2.5 Position Quality Evaluation

8.2.5.1 Position Deviation Metric

In order to follow G5 and G6, the labels should be placed along P. We have already given the procedure that generates the candidate label positions with their centers near the points q_j. If an input polyline is more curved, as it is often the case when a border is following a natural feature (e.g., rivers, mountain ranges, etc.), it is not always possible to make a label placement at a certain position q_j. Therefore, our method allows increasing the number of candidate positions around the certain position q_j. These potential label placements are anchored at $\bar{q}_{jk}$. It might be that two labels specified by two locations $\bar{q}_{jk}$ and $\bar{q}_{j+1k}$ from two different sets V_j and V_{j+1} are too close to each other. Thus, we need a metric to quantify the deviation of label candidate positions from an even distribution, i.e. the deviation of each point in V_j from the centre point q_j. A function for this metric has the following form:

$$g_{\mathrm{PosDev}}(l) = 1 - \delta / D_{\max} \tag{8.3}$$

where δ is the length of the part of the polyline that is bounded by the points q_j and $\bar{q}_{jk}$. Figure 8.10a depicts an example of the function for $g_{\mathrm{PosDev}}(l)$. It is clear from the figure that the metric (8.3) gives the best quality when $\delta = 0.0$, with $k = 0$. The worst case of the metric $g_{\mathrm{PosDev}}(l) = 0.0$ can be obtained by $\delta = D_{\max}$.



(a) (b) (c)

Figure 8.10. Quality functions used in the metrics. (a) Position deviation metric. (b) Baseline offset metric. (c) Goodness of fit metric.

8.2.5.2 Baseline Offset Metric

One of the output values of the candidate-position generation method are the offsets from the centerline R. We called them *baseline offsets*. Since the values of $h_\mathrm{l}, h_\mathrm{r}$ represents the maximum distance between the R and the points of P, it is clear that placing the labels on the lines BL_l and BL_r (Figure 8.9) respectively can lead to overlapping of P with descenders or ascenders of the label characters. To avoid this problem, we propose an additional offset to the values of h_l and h_r. The underlying idea is simple. We need to translate the baseline some distance in a direction perpendicular to the centerline. This additional offset we denote as ε. Note that the value of ε should be chosen by taking into account the font size of the label and the thickness (stroke width) of the boundary line.

Let us define a measure to quantify the quality of the labels with the given baseline offsets h_l, h_r and additional offset ε as follows:

$$g_{\mathrm{BaseOffset}}(l) = 0.5 \cdot u(h_\mathrm{l} + \varepsilon) + 0.5 \cdot u(h_\mathrm{r} + \varepsilon) \tag{8.4}$$

where function u is defined as:

$$u(\beta) = \begin{cases} 0, & \beta < B_{\min} \\ 1 - \dfrac{\beta - B_{\min}}{B_{\max} - B_{\min}}, & \beta \in [B_{\min}, B_{\max}]. \\ 0, & \beta > B_{\max} \end{cases} \tag{8.5}$$

where $B_{\min}, B_{\max}$ are minimum and maximum allowed offset values. Function (8.5) (Figure 8.10b) yields a value of 0.0 for the case when the distance β between the baseline of a label and the centerline is less than $B_{\min}$ or greater than $B_{\max}$, and a value of 1.0 when $\beta = B_{\min}$. It means that labels which have the distance from the centerline in the range $[B_{\min}, B_{\max}]$ are all acceptable. Note that the closer the label to the centerline is better. Parameter $B_{\max}$ defines the upper limit above which the label-feature association becomes unclear. The metric (8.4) scores how well the requirement of G4 is met. $B_{\min}$ and $B_{\max}$ are the control parameters and should be adjusted by the user.

8.2.5.3 Goodness of Fit Metric

We used the method of least squares in the procedure of candidate-position generation. Hence, we can calculate the coefficient of determination that equals the square of the correlation coefficient between the observed (polyline points) and modeled (centerline) data values for the case of a simple regression model. The coefficient of determination is a statistical characteristic that provides us with some information about the goodness of fit of a model. In our case it measures how well the centerline R locally approximates the polyline P (Figure 8.8). The coefficient of determination has values in the range [0,1], where a value of 1.0 indicates that the centerline fits the polyline perfectly. For instance, this case can be observed when all points, which have been used for the construction of R (see Phase II step 2 or Phase III step 4, respectively), lie on one line segment of P. Let us construct a metric that operates with the coefficient of determination. For this purpose we use an appropriate fading function as opposed to the metrics stated above. We also define a threshold to the value of the coefficient of determination, denoted as $C_{threshold}$. Figure 8.10c depicts an example of such a fading function. It can be seen from Figure 10c that the highest value is obtained if the goodness of fit is equal to the threshold value or higher. The quality of the fitting to the polyline deteriorates when the goodness of fit approaches 0. As a fading function we chose the following

$$g_{\text{GoodnessOfFit}}\left(l\right)=\begin{cases}1, & \gamma > C_{\text{threshold}}\\ 1-\left(\cos\left(\dfrac{\pi\cdot\gamma}{C_{\text{threshold}}}\right)+1\right)/2, & \gamma \le C_{\text{threshold}}\end{cases} \tag{8.6}$$

where γ is the value of the coefficient of determination. Note that metric (8.6) corresponds to G2.

8.2.5.4 Metric of Horizontal Alignment

This metric considers the cartographic guideline G7, which says “horizontally aligned labels are preferred to vertical ones”. In other words, the text should be as near to “reader normal” as possible (Wood, 2000). Therefore, we can determine the corresponding metric as follows:

$$g_{\text{HorizAlign}}\left(l\right)=1-\frac{\alpha}{90} \tag{8.7}$$

where α is the angle between the horizontal axis and the centerline which defines the orientation of the label l. The angle α is measured in degrees. Remember also, that α is one of the output values of the candidate-position generation procedure. Metric (8.7) is designed to yield a value of 1.0 for $\alpha = 0$. This is a case when the placement of a label is horizontal.

8.3 Experiments

In this section we provide some results of the experiments that we carried out to test the presented algorithm. We first describe our experimental methodology. Then we present the diagrams for performance and labelling quality measurements. We finish this section with sample maps generated with our algorithm.

8.3.1 Datasets, Implementation and Experimental Methodology

We have implemented a version of the proposed algorithm on top of the platform for publishing spatial data to the web. This platform is called *MapSurfer.NET* and written in C#. A machine for the experiments was equipped with an Intel® Core™ i5-2500 CPU @ 3.30 GHz and running Windows 7 Professional x64 with 8GB installed memory. A runtime execution environment of our test application was .NET Framework 4.5 (x64).

We performed our experiments on a dataset that represents geo-data provided by the *OpenStreetMap* project (OSM), one of the most promising crowd-sourced projects (Haklay and Weber, 2008; Ramm et al., 2010). For our experiments we choose a country with almost "complete" data for administrative divisions. This country is Italy. The sample dataset has been limited to a region that is located within the bounding box: 41.836501° N, 12.436859° E - 41.948695° N, 12.626374° E. We extracted all boundaries of municipalities from the OSM dataset and selected only the areal features with tag value of "admin_level=9" which is used in the region of interest to define administrative subdivisions in Rome. Then, we converted them to a format that has two additional attributes such as name_left and name_right. These attributes define the label content for the left and the right side of a polyline respectively.

The input parameters of our algorithm $S, D_{\max}, D_{\text{step}}, B_{\min}$ and $B_{\max}$ are measured in map units which are pixels in our tests. Additionally, in our implementation, we used a quality threshold parameter Q_{T} . This parameter allows controlling and eliminating candidate label positions that correspond to poor and sloppy label placement. These potential label positions are considered as unacceptable and omitted from the *position selection* procedure. Parameter Q_{T} takes values in the range [0,1], where a value of 1.0 corresponds to an ideal case. In the tests we used $D_{\text{step}} = 1$ and chose the value of the parameter K (see Phase II step 1) sequentially from the set $\{0.5, 0.55, 0.6, 0.65, 0.7, 0.75\}$ until a label placement was found. Next, to evaluate each label position we used the function (8.2) with two different sets of parameters, namely:

$$
\begin{aligned}
F_1(l) &= 0.3 \cdot g_{\mathrm{PosDev}}(l) + 0.1 \cdot g_{\mathrm{BaseOffset}}(l) + 0.5 \cdot g_{\mathrm{GoodnessOfFit}}(l) + \\
&\quad +0.1 \cdot g_{\mathrm{HorizAlign}}(l) \\
F_2(l) &= 0.1 \cdot g_{\mathrm{PosDev}}(l) + 0.5 \cdot g_{\mathrm{BaseOffset}}(l) + 0.3 \cdot g_{\mathrm{GoodnessOfFit}}(l) + \\
&\quad +0.1 \cdot g_{\mathrm{HorizAlign}}(l)
\end{aligned}
\tag{8.8}
$$

In function F_1 we incorporate the preference of $g_{\mathrm{GoodnessOfFit}}$ and nearly neglect the influence of $g_{\mathrm{BaseOffset}}$ and $g_{\mathrm{HorizAlign}}$. In function F_2 we set the lowest priority to g_{PosDev} and $g_{\mathrm{HorizAlign}}$. We shifted the importance of $g_{\mathrm{GoodnessOfFit}}$ to the second place, whereas metric $g_{\mathrm{BaseOffset}}$ received the highest priority.

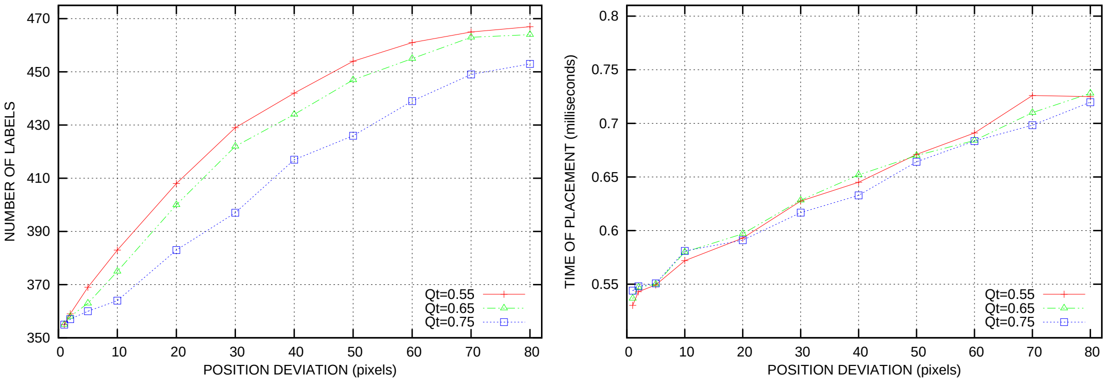
Besides four weights in the quality function, our algorithm consists of several input parameters that are defined by a user. Generally speaking, the number of parameters that are needed to be set by a user can be reduced in a certain implementation of the method in a GIS application. More exactly, the weights m_1, m_2, m_3, m_4 and the parameters $D_{\max}, D_{\mathrm{step}}, B_{\min}, B_{\max}$, and Q_{T} need only be defined once according to the preferences of a cartographer or predefined by a software developer. Next, the parameter S can be computed in the runtime by taking into account some minimum permissible distance and the length of the labels to be placed. Note, however, that the availability of the parameters in a user interface gives to the cartographer much more freedom in labelling. The exemplary parameters given below were chosen for the case of web-mapping for OSM data.

8.3.2 Performance and Visualization Results

In the first set of experiments we used quality function F_1 and tried to find out how the success rate (the number of potential label locations) decreases as we increase the quality threshold Q_{T} and the position deviation $D_{\max}$, and how much time the labelling takes. We set the input parameters to $S = 400$, $B_{\min} = 2$, $B_{\max} = 8$. Then, taking into account the value of S and the length of each of the 46 polylines in the tested region, we calculated the maximum possible number of labels. This number was $m = 493$. It is worth noting that from each set V_j of the candidate positions $\bar{q}_{jk}$, we choose only one candidate. In Figure 8.11a we present the results of the experiment. It can be seen that the algorithm is able to place the labels in 95% of the desired positions ($m = 493$) with $Q_{\mathrm{T}} = 0.55$ and $D_{\max} = 80$. We can observe 75% of the maximum possible number of labels in the case when the quality threshold is higher, namely $Q_{\mathrm{T}} = 0.75$ and $D_{\max} = 1$. Therefore, we conclude that enlargement of the search space and a lowered quality threshold results in a higher rate of labeled positions. Furthermore, in order to determine the influence of the search space on the algorithm's runtime, we provide the time measurements of the tests where we varied $D_{\max}$. Figure 8.11b illustrates linear dependence. Moreover, according to this figure, our algorithm is able to find one label position in 0.007602 seconds. Note that such performance makes the algorithm appropriate for usage in interactive and dynamic labelling (Been et al., 2006; Mote, 2007).

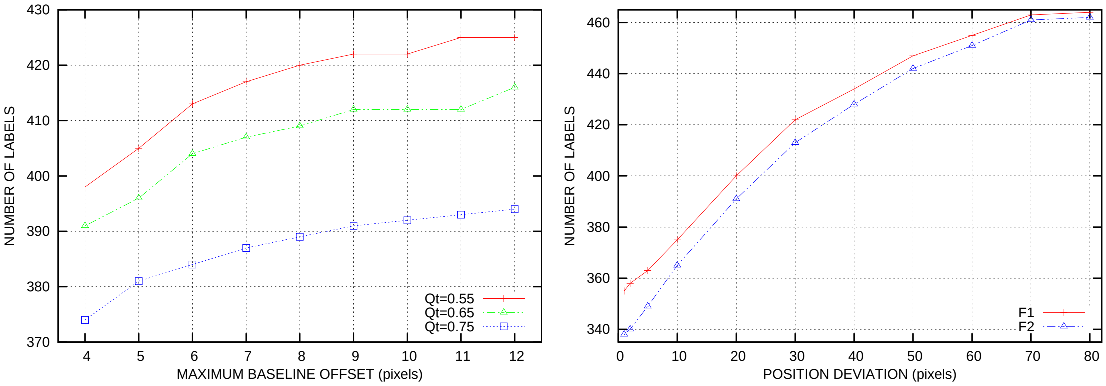
In another test, we fixed the position deviation value $D_{\text{max}} = 25$ and ran our algorithm several times by varying B_{max}. The results of the tests are shown in Figure 8.12a. The results illustrate the ability of the algorithm to increase the percentage of placements by increasing the maximum permissible distance between two coupled labels (n_{l} and n_{r}) on either side of the polyline. This possibility comes in handy in case of labelling extremely curved parts of a polyline.

Finally, we evaluated the dependence of the number of placed labels on the type of the quality function. We ran the same test for both functions F_1 and F_2. The results



(s) (b)

Figure 8.11. Experimental results for different Q_{T} values. (a) Dependence of the number of label placements on D_{max}. (b) Dependence of the run time on D_{max}. The Y-axis defines time which is needed to find one label placement.



(a) (b)

Figure 8.12. Experimental results. (a) Dependence of the number of label placements on B_{max} when $B_{\text{min}} = 2$. Tested function is F_1. (b) Comparison of two quality functions F_1 and F_2 when $B_{\text{min}} = 2, B_{\text{max}} = 8$.

presented in Figure 8.12b show that the algorithm places more labels with function F_1 than with F_2. However, the number of labels is almost the same with higher values of $D_{\max}$.

In order to demonstrate that our algorithm is able to generate legible and cartographically plausible label placements, we prepared two sample maps (Figures 8.12 and 8.13). We used function F_1 for type placement in both maps. Figure 8.13 depicts a map which was labeled using a small number of candidate positions and a high value of Q_{T}. Figure 8.14 shows the same map region as in Figure 8.13, but the labelling was performed with a different set of input parameters. Namely, the number of candidate positions was much greater and the requirements to the quality of a label position were lower. As a result, the algorithm placed 2.86 times more labels when using the second set of parameters.

As it can be observed from Figure 8.14 (see red marks), in some cases our algorithm places labels which overlap the corresponding polyline which has a sinous shape. This fact contradicts G9. This inability of the algorithm can be overcome by performing an additional post-processing step. To check whether a polyline and its label intersect, we can utilize the algorithm by Shamos and Hoey (1976) for reporting intersections between two sets of line segments. The polyline composes the first set of line segments, whereas eight line segments (8 for each pair of labels) bounding the label comprise the second set.

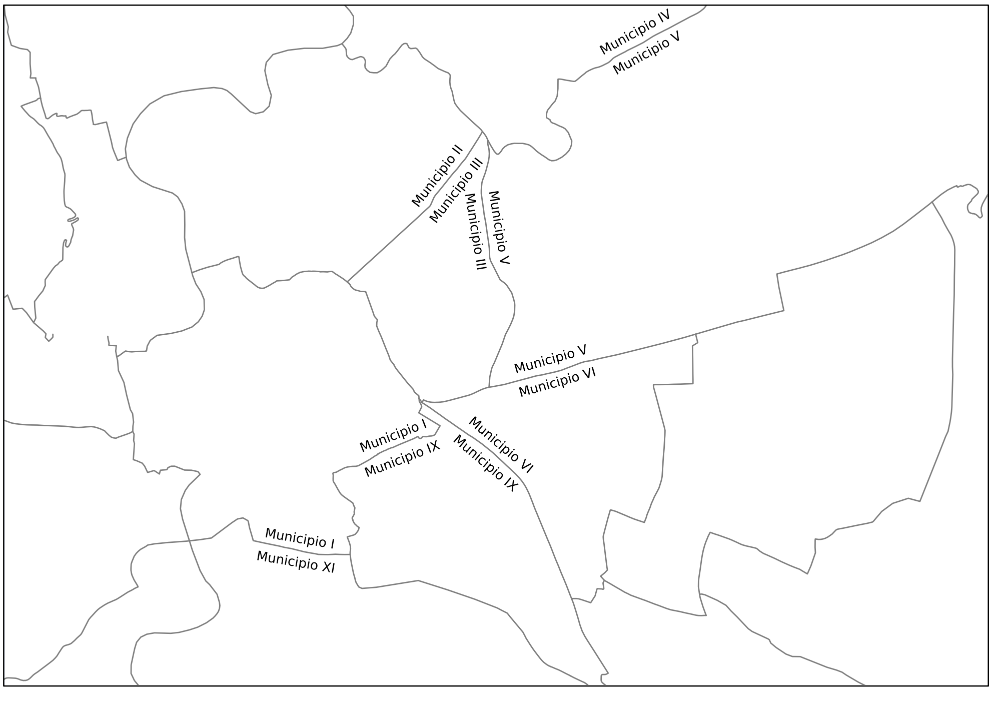


Figure 8.13. Labelling of the municipality boundaries of Rome (7 labels). The input parameters are $S = 400, D_{\max} = 1, B_{\min} = 2, B_{\max} = 4, Q_{\mathrm{T}} = 0.75$.

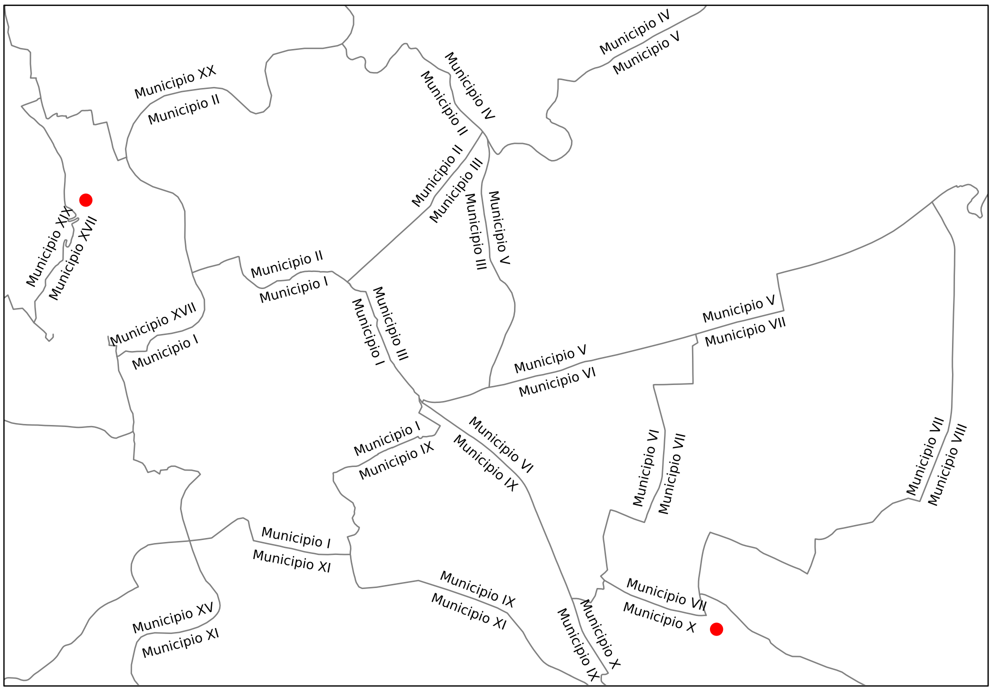


Figure 8.14. Labelling of the municipality boundaries of Rome (20 labels). The input parameters are $S = 400, D_{\text{max}} = 100, B_{\text{min}} = 2, B_{\text{max}} = 10, Q_{\text{T}} = 0.55$.

For the sake of the performance this check should be done only once after all potential labels are generated, i.e. the second set of line segments consists of $8N_{\text{total}}$ elements, where N_{total} is the total number of label pairs for the polyline. Note that our implementation currently does not take this extra step.

Figure 8.15 illustrates a part of a map generated by MapSurfer.NET. In contrast to the map in Figure 8.14, this map contains different types of designation such as points (e.g., settlements, motorway shields and peaks), curved lines (e.g., streets, rivers, boundaries) and areas (e.g., parks, lakes) labelling. This map demonstrates the possibility of using our algorithm as a part of a more general labelling algorithm (Edmondson et al., 1997).

A set of maps involving pairwise line labelling of boundaries are available online through a web map tile service (WMTS; García et al., 2012) on the *OpenMapSurfer* (2014) web page. On this page the layers “OSM Roads”, “OSM admin Boundaries” demonstrate the output of the algorithm on the OSM dataset for the whole globe.

Figure 8.15. A sample map containing lettering of administrative boundaries together with other feature types such as roads, railways, districts, parks, etc. Projection: spherical Mercator (EPSG:3857). Data source: © OpenStreetMap contributors (2013, data licensed under ODbL).

8.3 Conclusions and Future Work

In this paper we have introduced a new efficient and easily configurable algorithm for performing visually plausible and functional pairwise labelling of lines presenting geographic boundaries. Our algorithm achieves two goals; it generates candidate positions and evaluates their quality according to the predefined set of cartographic guidelines for line labelling.

The results of our experiments on a real-world dataset showed that our algorithm is able to find the candidates in 95% of all possible positions with a certain set of input parameters. The runtime measurements confirmed the high performance of the algorithm. Another advantage of the algorithm is that the generated candidate positions and the quality function can be used in a general map labelling algorithm (Edmondson et al.,

1996) that label all feature types (e.g., points, lines and polygons) simultaneously. More precisely, the quality function can potentially be used as a component for a comprehensive quality function (van Dijk et al., 2002; Rylov and Reimer, 2014a for point-features) which is employed by a combinatorial optimization algorithm (Christensen et al., 1995) to find the globally best and optimal label placement. We also believe that our algorithm can be easily reproduced and embedded in one of the commercial or open source GIS toolkits.

It remains an open problem how to perform pairwise labelling of boundary lines using curved text as depicted in Figure 8.2, which is often more preferable. This task can be accomplished by exploiting a curve fitting procedure. Note that it will require a new method for candidate positions generation and the construction of another quality function. Moreover, both parts of the algorithm should be based on an adopted list of cartographic guidelines. We think that some parts of our algorithm can be used as a baseline for the construction of a new method.

In conclusion, we sincerely hope that our approach advances the development of more robust and efficient algorithms for labelling geographic boundaries.

Acknowledgements

Maxim Rylov is grateful to DAAD (Deutscher Akademischer Austauschdienst, German Academic Exchange Service) for awarding DAAD-Scholarship to pursue his doctoral degree. Moreover, the authors would like to thank the anonymous reviewers for their constructive comments and suggestions.

References

Barrault, M. 2001. “A Methodology for Placement and Evaluation of Area Map Labels.” *Computers, Environment and Urban Systems* 25(1): 33-52. doi: 10.1016/S0198-9715(00)00039-9

Barrault, M., and F. Lecordix. 1995. “An Automated System for Linear Feature Name Placement which Complies with Cartographic Quality Criteria.” In *Proc. AutoCarto 12:* 321-330. ACSM/ASPRS, Bethesda.

Basoglu, U. 1982. “A new approach to automated name placement.” In *Proc. AutoCarto 5*: 103-112.

Been, K., E. Daiches, and C. Yap. 2006. “Dynamic Map Labeling.” *IEEE Transactions on Visualization and Computer Graphics* 12(5): 773-780. doi: 10.1109/TVCG.2006.136

Brewer, C.A. 2005. Designing Better Maps: A Guide for GIS Users, ESRI Press, Redlands CA.

Chatterjee, S., and A. Hadi. S. 2006. "Simple Linear Regression." In *Regression Analysis by Example*, 4th ed. New Jersey: John Wiley & Sons.

Chirié, F. 2000. "Automated Name Placement With High Cartographic Quality: City Street Maps." *Cartography and Geographic Information Science* 27(2): 101-110. doi: 10.1559/152304000783547902

Christensen, J., J. Marks, and S.M. Shieber. 1995. "An Empirical Study of Algorithms for Point-Feature Label Placement." *ACM Transactions on Graphics* 14(3): 203-232. doi: 10.1145/212332.212334

Defense Mapping Agency. 1973. Series 1501 AIR, Sheet ND 48-12, Virachey, Cambodia;Vietnam, 1:250 000, Edition 3. Scan provided by University of Texas, Perry-Castañeda Library Map Collection. Accessed January 20, 2014. http://www.lib.utexas.edu/maps/jog/vietnam/nd-4812_geo.pdf.

Edmondson, S., J. Christensen, J. Marks, and S. Shieber. 1996. "A General Cartographic Labeling Algorithm." *Cartographica* 33(4): 13-23. doi: 10.3138/U3N2-6363-130N-H870

García, R., J.P. de Castro, E. Verdú, M.J. Verdú, and L.M. Regueras. 2012. "Web Map Tile Services for Spatial Data Infrastructures: Management and Optimization." In *Cartography - A Tool for Spatial Analysis*. ed. C. Bateira. 25-48. doi: 10.5772/46129

Haklay, M., and P. Weber. 2008. "OpenStreetMap – User Generated Street Map", IEEE Pervasive Computing 7(4): 12-18. doi: 10.1109/MPRV.2008.80

Hirsch, S.A. 1982. "An Algorithm for Automatic Name Placement around Point Data." *American Cartographer* 9(1): 5-17. doi: 10.1559/152304082783948367

Imhof, E. 1962. "Die Anordnung der Namen in der Karte." In *International Yearbook of Cartography*, 93-129, Bonn-Bad Godesberg: Kirschbaum.

Imhof, E. 1975. "Positioning names on maps." *American Cartographer* 2(2): 128-144. doi: 10.1559/152304075784313304

Kakoulis, K. G. and I. G. Tollis., 2003. "A Unified Approach to Automatic Label Placement." *International Journal of Computational Geometry and Applications* 13(1): 23-59. doi: 10.1142/S0218195903001062

Kern, J.R., and C.A. Brewer. 2008. "Automation and the Map Label Placement Problem: A Comparison of Two GIS Implementations of Label Placement." *Cartographic Perspectives* 60, 22-45.

MapSurfer.NET. 2014. A framework for map styling and rendering. Available at http://openmapsurfer.uni-hd.de/Framework/info.html

Mote, K. 2007. "Fast Point-Feature Label Placement for Dynamic Visualizations." *Information Visualization* 6(4): 249-260. doi: 10.1057/PALGRAVE.IVS.9500163

National Imagery and Mapping Agency. 1962. Series Z901, Sheet S6/13 SW-4B, Matadi, 1:12 500, Edition 2 – AMS, Prepared by the Army Map Service (TV). Scan provided by University of Texas, Perry-Castañeda Library Map Collection. Accessed January 20, 2014. http://www.lib.utexas.edu/maps/africa/txu-oclc-55853021-matadi-1959.jpg

OpenMapSurfer, 2014. (a web map service provided by Heidelberg University; accessed July 4, 2014). Available at http://openmapsurfer.uni-hd.de.

Rabello, R.L., G.R. Mauri, G.M. Ribeiro, and L.A.N. Lorena. 2014. "A Clustering Search Metaheuristic for the Point-Feature Cartographic Label Placement Problem." *European Journal of Operational Research* 234:802–808. doi: 10.1016/j.ejor.2013.10.021

Ramm, F., J. Topf, and S. Chilton. 2010. *OpenStreetMap: Using and Enhancing the Free Map of the World*. Cambridge: UIT Cambridge.

Regnauld, N, S. Lessware, C. Wesson, and P. Martin. 2013. "Deriving Products from a Multiple Resolution Database using Automated Generalisation at Ordnance Survey." In Proc. 26th International Cartographic Conference, August 25-30, Dresden, Germany.

Revell, P., N. Regnauld, and G. Bulbrooke. 2011. "OS Vectormap District: Automated Generalisation, Text Placement and Conflation in Support of Making Public Data Public." In Proc. 25th International Cartographic Conference, July 3-8, Paris, France.

Rylov, M.A., and Reimer, A.W. 2014a. "A Comprehensive Multi-Criteria Model for High Cartographic Quality Point-Feature Label Placement." *Cartographica* 49(1): 52-68. doi: 10.3138/carto.49.1.2137.

Rylov, M.A., and Reimer, A.W. 2014b. "A Practical Algorithm for the External Annotation of Area Features." *Cartographic Journal*. doi: 10.1179/1743277414Y.0000000091

Rylov, M.A., and Reimer, A.W. 2014c. "Improving Label Placement Quality by considering Basemap Detail with a Raster-Based Approach." *GeoInformatica*. doi: 10.1007/s10707-014-0214-6

van Dijk, S., M.J. van Kreveld, T.W. Strijk, and A. Wolff. 2002. "Towards an Evaluation of Quality for Names Placement Methods." *International Journal of Geographical Information Science* 16(7): 641-661. doi: 10.1080/13658810210138742 van Kreveld, M.J., T.W. Strijk, and A. Wolff. 1999. "Point Labeling with Sliding Labels." *Computational Geometry: Theory and Applications* 13: 21-47. doi: 10.1016/S0925-7721(99)00005-X

van Roessel, W. 1989. "An Algorithm for Locating Candidate Labeling Boxes Within a Polygon." *American Cartographer* 16(3): 201-209. doi: 10.1559/152304089783814034

Verner, O.V., R.L. Wainwright, and D. Schoenefeld. 1997. "Placing Text Labels on Maps and Diagrams using Genetic Algorithms with Masking." *INFORMS Journal on Computing* 9(3): 266-275. doi: 10.1287/ijoc.9.3.266

Wolff, A., L. Knipping, M. van Kreveld, T. Strijk, and P. K. Agarwal. 2001. "A Simple and Efficient Algorithm for High-Quality Line Labeling." *Innovations in GIS VII: GeoComputation* 11: 147-159.

Wolff, A., and T. Strijk, 2009. "A map labeling bibliography." Available at http://i11www.iti.uni-karlsruhe.de/~awolff/map-labeling/bibliography/.

Wood, C.H. 1994. "Effects of Brightness Difference on Specific Map-Analysis Tasks: An Empirical Analysis." *Cartography and Geographic Information Science* 21(1): 15-30. doi: 10.1559/152304094782564019

Yamamoto, M., G. Camara, and L.A.N. Lorena. 2002. "Tabu Search Heuristic for Point-Feature Cartographic Label Placement." *Geoinformatica* 6(1): 77-90. doi: 10.1023/A:1013720231747

Yoeli, P. 1972. "The Logic of Automated Map Lettering." *Cartographic Journal* 9(2): 99-108. doi: 10.1179/000870472787352505

Declarations of Authorship

Declaration of Authorship

Paper Title: A Comprehensive Multi-Criteria Model for High Cartographic Quality Point-Feature Label Placement.

Appeared in: Rylov, M. A., Reimer, A. W. 2014. “A Comprehensive Multi-Criteria Model for High Cartographic Quality Point-Feature Label Placement.” *Cartographica* 49(1): 52-68. doi: 10.3138/carto.49.1.2137

Contribution Statement:

Maxim Rylov has developed and implemented the presented model himself. Moreover, he has written the major part of the manuscript. Andreas Reimer has contributed to the paper by sharing his knowledge in the field of cartography, continuing discussions about the used methods and the results of the experiments. One of his notable contributions was a suggestion to add a table that shows how many cartographic requirements for point-feature labelling has been considered in previous research works. Furthermore, an extensive proof-reading by the co-author has led to the substantial improvement of the research paper.

The estimated percentage of the dissertation author’s contribution is 80%.

Date	Name	Signature
2.9.2014	Andreas Reimer	

Declaration of Authorship

Paper Title: A Practical Algorithm for the External Annotation of Area Features.

Appeared in: Rylov, M., Reimer, A. 2013. "A practical algorithm for the external annotation of area features." *Cartographic Journal* 51(3). doi: 10.1179/1743277414Y.0000000091

Contribution Statement:

Maxim Rylov has devised the algorithm and written the major part of the manuscript. Andreas Reimer has supported this publication through continuing discussions, giving suggestions regarding cartographic aspects of labelling areal features and preparing some illustrations. He also made a significant contribution by describing the proposed quality evaluation function. Furthermore, an extensive proof-reading by the co-author has led to the substantial improvement of the research paper.

The estimated percentage of the dissertation author's contribution is 80%.

Date	Name	Signature
	A. Zipf	A Zi
2.9.2014	Andreas Reimer	A.

Declaration of Authorship

Paper Title: Improving Label Placement Quality by considering Basemap Detail with a Raster-Based Approach.

Appeared in: Rylov, M., Reimer, A. 2014. "Improving Label Placement Quality by considering Basemap Detail with a Raster-Based Approach." *GeoInformatica.* doi: 10.1007/s10707-014-0214-6

Contribution Statement:

Maxim Rylov has developed the method and has written the major part of the paper. Andreas Reimer has supported this publication through continuing discussions, sharing his expertise in cartography and thoroughly helped to process the results of the experiments. Furthermore, an extensive proof-reading by the co-author has led to the substantial improvement of the publication.

The estimated percentage of the dissertation author's contribution is 75%.

Date	Name	Signature
	A. Zipf	A. Zipf
2.9.2014	Andreas Reimer	

Declaration of Authorship

Paper Title: Pairwise Line Labelling of Geographic Boundaries: An Efficient and Practical Algorithm.

Appeared in: Rylov, M., Reimer, A. 2014. "A Pairwise Line Labelling of Geographic Boundaries: An Efficient and Practical Algorithm." *Cartographic Perspectives*, (submitted).

Contribution Statement:

Maxim Rylov has developed the algorithm and written the major part of the manuscript. Andreas Reimer has contributed to the paper through continuing discussions about the cartographic guidelines, automated methods and the results of the experiments. Furthermore, an extensive proof-reading by the co-author has led to the substantial improvement of the research paper.

The estimated percentage of the dissertation author's contribution is 80%.

Date	Name	Signature
	A. Zipf	A. Zipf
2.9.2014	Andreas Reimer	A. Rei

Eidesstattliche Versicherung gemäß §8 der Promotionsordnung der Naturwissenschaftlich-Mathematischen Gesamtfakultät der Universität Heidelberg

1. Bei der eingereichten Dissertation zu dem Thema “*Models and Methods for Cartographic Label Placement*” handelt es sich um meine eigenständig erbrachte Leistung.

2. Ich habe nur die angegebenen Quellen und Hilfsmittel benutzt und mich keiner unzulässigen Hilfe Dritter bedient. Insbesondere habe ich wörtlich oder sinngemäß aus anderen Werken übernommene Inhalte als solche kenntlich gemacht.

3. Die Arbeit order Teile davon habe ich bislang nicht an einer Hochschule des In- oder Auslands als Bestandteil einer Prüfungs- oder Qualifikationsleistung vorgelegt.

4. Die Richtigkeit der vorstehenden Erklärungen bestätige ich.

5. Die Bedeutung der eidesstattlichen Versicherung und die strafrechtlichen Folgen einer unrichtigen oder unvollständigen eidesstattlichen Versicherung sind mir bekannt.

Ich versichere an Eides statt, dass ich nach bestem Wissen die reine Wahrheit erklärt und nichts verschwiegen habe.

Heidelberg, 1 Oktober 2014

Maksim Rylov