Towards an Evaluation of Quality for Names Placement Methods

Steven van Dijk* Marc van Kreveld* Tycho Strijk[†] Alexander Wolff[‡]

Abstract

The cartographic labeling problem is the problem of placing text on a map. This includes the positioning of the labels, and determining the shape in the case of line and area feature labels. There are many rules and customs that describe aspects of good label placement, like readability and clear association. This paper gives a classification of most label placement rules, and formalizes them into a function that can serve as a quality measure for label placement. If such a function is implemented, it allows to compare the output of different label placement programs. We give a simple and a more refined example of the quality function.

1 Introduction

For a human cartographer label placement is a labor intensive task that can take up to 50% of the total map design time. This explains why there have been many attempts to automate name placement, mostly in the last two decades, see the bibliography of papers on map labeling maintained by Wolff and Strijk (1996). Automated label placement requires that the standard guidelines for label placement be formalized. The simplest of all rules to follow is that names shouldn't overlap. Most automated label placement methods described in the literature comply with this rule; however, only a few methods take into account other aspects such as aesthetics and avoiding ambiguity.

In this paper we formalize and classify most requirements that are relevant to the positioning of names on a map. We develop a *quality function* for label placement methods that measures how well a method places labels on a given map. Such a function is useful in many ways. Firstly, it helps us to understand what contributes to good label placement in general. Secondly, it helps us to develop label placement software that takes into account the various aspects of label placement. High-quality cartographic label placement involves many, often conflicting factors. This makes it difficult to develop software that takes into account all aspects simultaneously, and is efficient at the same time. This brings us to the third use of the quality function: it provides a way to compare different label placement methods. From a computational point of view, it is much easier and faster to measure quality than to optimize it.

The remainder of this paper is organized as follows. In the next section we will globally discuss the criteria for high-quality map labeling as described in the cartographic literature, and list those that are currently considered by automated label placement methods. In Section 3 we develop a framework for a quality function in which nearly all criteria can be put. This includes specifying the map elements, assumptions on labels, and the form of the quality function. In Section 4 we take a more detailed look at the criteria from literature to see how they fit into our framework. In Section 5 we list those criteria that were not included to keep the model simple and suggest

*Department of Computer Science, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands {steven, marc}@cs.uu.nl

[†]ASM Lithography, Veldhoven, The Netherlands t.strijk@freeler.nl

[‡]Institut für Mathematik und Informatik, Ernst-Moritz-Arndt-Universität Greifswald, Jahnstraße 15a, D-17489 Greifswald, Germany awolff@mail.uni-greifswald.de

ways to extend the framework. In Section 6 we make the quality function more concrete by giving an example of a fully operationalized quality function that fits into our framework. We both give a simple version and a more extended version, showing that most criteria can be captured quantitatively. Section 7 concludes the paper.

2 Aspects of letterings maps

A map—on paper or on a screen—consists of the representation of a number of geographic objects and their annotation. Geographic objects are objects that have real world coordinates associated to them, and when represented on a map they can be zero-, one-, or two-dimensional. They can also be symbolized, or composite objects. Annotation consists of labels, that is, text or symbols associated to map objects, and miscellaneous map objects like title and legend. The main characteristic of annotation is that its position on the map is not determined directly by coordinates in the real world.

2.1 Cartographic criteria

The cartographers Imhof (1975), Alinhac (1962), and Yoeli (1972) have each listed a number of requirements for high-quality labeling. These requirements have been summarized in various textbooks and surveys, e.g. by Dent (1996) and Robinson et al. (1995). Here we briefly list the major high-level rules again.

Legibility: Influenced by font size, color (contrast with background), overlap with other labels and features.

Aesthetics: Influenced by the choice of font, shape of text, clustering (clutter), accidental regularity in text.

Harmony: Influenced by the choice of typeface, type family, and color for all labeled map objects. It is considered good practice to select one typeface, but allow several variants of a type family, e.g. allow Times roman and italic, variation in weight (light, medium, bold), and a small number of font sizes.

Unambiguity: Involves avoiding text close to objects to which it does not correspond. Harmony may help to resolve ambiguities. If river names are blue and city labels black, a river name won't be mistaken for a city name, even if it is the text closest to the city.

Not disturbing the map contents: Text must sometimes be placed on top of other objects, but should not cover important information or relevant details of the map. Note that here the emphasis is on the map background while in the criterion legibility emphasis is on the label.

Suggesting position, orientation, shape, and hierarchy: One example of this aspect is that larger cities should have their name in a larger font than smaller cities, and that coastal towns should have their name in the sea. Other examples are map features with indeterminate boundaries, features whose boundary is not explicitly shown, and composite features like groups of islands.

Notice that not all aspects of these rules are relevant to the *placement* of text, for instance the issues listed under "Harmony" and some of the issues listed under "Aesthetics". These issues will not be evaluated when we discuss our quality function, since we restrict ourselves to evaluating label placement.

2.2 Criteria in automated names placement research and software

The rules used in methods for automated label placement are usually much more 'down to earth' than the cartographer's requirements. Most attention has been given to labeling point features. In many cases, the only requirement is that labels may not overlap and that the bounding box of the text touches the point to be labeled, to make sure that the point and the label associate with each other.

Yoeli (1972) was the first to study automated label placement, and incorporated position preferences in his algorithm. Later many others followed this. Hirsch (1982) used buffer circles around points. By forbidding other labels to intersect these buffers, he prevented ambiguity. Ahn and Freeman (1984) developed a name-placement system called AUTONAP that includes area labeling. AUTONAP adapts the label shape to the so-called skeleton of an area. Langran and Poiker (1986) combine label placement and point selection by removing the least important points in overcrowded map regions so that the remaining labels can be placed without overlap.

Jones (1989) tried to avoid disturbing the map contents by placing a grid of pixels on the map. Different feature classes have an overlapping priority. For each pixel the highest priority feature is determined that overlaps the pixel. For each label position the amount of overlap is the sum of the priorities of the pixels covered by the label. Labels are placed preferably in positions with a small amount of overlap. Cook and Jones (1990) describe a Prolog rule that determines whether a label of a city is placed on the same side of a river or boundary as the city symbol. Another Prolog rule they describe is to halve the distance of a label to its point if the point lies in a crowded area. This is done in order to prevent ambiguity. Doerschler and Freeman (1992) also describe a rule-based approach. One rule they use in order to avoid ambiguity is that route numbers may not be placed at the intersection of two roads.

Edmondson et al. (1997) implemented a simulated-annealing algorithm that gives penalties for label-label and feature-label overlaps. They, too, introduced a metric to measure how well a line label is placed with respect to the line. Pinto and Freeman (1996) introduce a measure that evaluates how well an area label is placed in five respects. Barrault (1997) and Wolff et al. (1999) both give algorithms for high-quality line labeling where several different criteria are used.

Preuß (1998) proposed an objective function for point labeling as part of an evolution strategies approach that incorporates the following elements. Firstly, the label of a point feature must keep a certain distance to any other point features and labels. Secondly, for every point feature the labels that do not belong to it must keep a certain distance. Thirdly, there is a minimum and a maximum for the distance between a point feature and its own label. The objective function of Preuß also takes into account general preferential positions of labels relative to their point features. In addition he computes the local density around each point feature and uses these densities for a function that gives a penalty for clustered labels and big variance of local density.

There are three commercial software packages that perform high-quality label placement in many cases. These are Maplex (ESRI 1998), MapText (1999), and EverName (EverMap 1999). They incorporate many different criteria in their label placement algorithms, and allow the user to specify personal preferences as well. Another assessment of labeling quality in many different respects is given in the thesis of Barrault (1998). His research is closely related to the study presented in this paper.

3 A specification for name placement quality

In this section we will work towards the development of a function that measures how well a label placement algorithm performed its task on a given input. This function will be called the *quality function*. We'll first specify what constitutes a map, what labeling restrictions we impose a priori, and what we expect a label placement algorithm to get as input and return as output.

The result obtained from the quality function should be easy to interpret. The easiest type of output is a *value* for the quality. This value should be higher when labels are placed better. Such an output allows comparing two different labelings of the same base map. In our opinion, the value may—at best—be interpreted on an ordinal scale, since statements about exactly how much one labeling is better than another will generally be meaningless. Defining an ordering in labeling quality seems ambitious already. The goal is to define an ordering for all cases where one labeling is clearly better than another. If two human cartographers could decide differently on which label placement is better, then the ordering induced by the quality function can follow only one judgement, of course.

Our quality function should have the properties simplicity, relevance, tractability, and understandability. *Simplicity* is a general asset of a model, which the quality function is. The type of *relevance* we expect is that the quality function really gives an ordering by quality of different labelings of the same map. *Tractability* is necessary since the quality function should be a tool to be implemented which allows automated comparisons. The *understandability* of the model is important since it should be adjustable to give better results. To this end, the quality function should depend only on a few parameters whose tuning has predictable effects on the result of the quality function.

3.1 Elements of the map

In order to formally deal with map labeling, we first need to specify what objects are present on a map. A map generally is a rectangular area with the following map objects that have a fixed position:

- Point features, which can be point symbols or other (pictorial) symbols.
- Linear features, which can be lines or curves, and may branch and join.
- Areal features.

Each of these features has a display style including aspects like color, pattern, and width for lines. Note that linear features can have many different meanings. They can represent roads or rivers, administrative boundaries, parallels of latitude, or contour lines.

A map usually contains other objects as well, objects without a fixed position in the real world:

- Title, legend, and other insets of the map.
- Text labels associated with any of the three types of features listed above.
- Leader symbols such as arrows that convey the association between a feature and its label.
- Diagrams and charts located inside or close to the features they are associated with.

All of these objects—both the ones with fixed position in the real world and the ones without—have an explicit position on the map when it is finished, and that map can be described purely by geometry (coordinates) and display style of its objects.

3.2 A-priori restrictions of label placement

In the following we list a number of restrictions of our map model. Among others these will affect the type and shape of labels that a label placement algorithm can generate if it adheres to our model. One reason to have such limitations beforehand is that the quality function to be developed need not deal with the most pathological situations. The advantage is that the quality function will give a better indication of the quality in the more normal cases. A second reason is that the quality function to be developed later can be simpler; simplicity is a desirable property for any model.

The first limitations we make are that we don't consider the map elements title and legend, and we don't deal with diagrams and charts on the map, and visual effects like hill shading. We also don't treat marginal annotation and the use of leader symbols. Secondly, any feature receives at most one label on the map. The other limitations deal with the allowed shape of labels. We assume that the text itself, the type, typeface, font, and color of the text is fixed for each feature to be labeled. Text is not split over lines. The text block is treated as a rectangular block, somewhat like a bounding box. Point labels are shapes that must align with parallels of latitude on the map. On large scale maps, the parallels are usually horizontal, and all point labels are axis-aligned rectangles. If the parallels are curved, then the rectangle is curved like the parallels. Line labels are text blocks that, intuitively, are “freely curved” rectangles. They are specified by a base curve and a height. For area labels we additionally allow spacing between individual characters of the text. Area labels are specified by a base curve and a height. Spacing is restricted to be equal between any two characters. So an area label consists of a set of curved rectangles (one for each character) on a common base curve, where the distance between consecutive rectangles is fixed.

3.3 Input and output of a label placement method

Before a label placement method can start, a number of choices about the map design must be made. For instance, what features to display, what colors and which map projection to use. Intuitively, the map objects have been drawn but no labels have been placed yet. However, which features should receive labels and in which font has been decided. Let $F = F^+ \cup F^-$ be the set of features, where F^+ and F^- are the sets of features that will and won't be labeled, respectively. Note that features in F^- must also be part of the input since they must be considered when placing labels of other features. Let L be the set of labels of F^+.

Thus we formalize the input to a label placement method by the set F^- of features that will not be labeled and the set P of pairs $\langle f, \ell \rangle$ where f is a feature in F^+ and ℓ the corresponding label in L. Each feature has the following attributes:

feature attributes:

- type (point, line or area)
- geometry (position, shape, orientation)
- display style (color, pattern, ...)
- priority

The priority is an attribute that captures how important a feature is for the intended map use. Each label on the map is modeled as follows:

label attributes:

- text
- geometry (position, shape, reading direction)
- display style (color, typeface, font, ...)
- priority

The priority of a label may be different from the priority of its feature, because sometimes the presence of a feature is more important than its label. The second attribute of a label is the part computed by a label placement method. If a label isn't placed, the geometry attribute states UNLABELED. We assume that the output of the label placement method consists of F^- and P and that the geometry attribute of each $\ell \in L$ is specified. Note that the output contains more information than the labeled map that will be drawn based on it, because from this map one cannot determine the priority of its features and labels, nor which labels were intended to be placed but couldn't. This information is necessary to evaluate the quality of the result of a label placement algorithm.

3.4 Form of the quality function

The quality function evaluates how well a label placement method performed its task. The quality function should reflect that the quality is low when labels are missing or poorly placed, especially if labels of high priority are involved. The input to our quality function consists of the output of the label placement method as specified above.

The quality function for label placement is a combination of the qualities of the individual label positions. This makes it possible to trace low quality to the labels that cause it. There are four aspects to the quality of label placement represented in our function. They form a break-down of the overall label placement quality into natural categories with only little overlap in meaning.

We outline the four categories and specify for each which parameters are needed to evaluate it. The first parameter is always the one whose quality is being assessed. Note that the partial quality functions also evaluate the corresponding feature and label priorities since we consider these to be attributes of the features and labels.

Aesthetics. This aspect represents the quality of the shape of the label itself and is not influenced by the label's position on the map, nor by other map features. The aesthetic quality of a label $\ell \in L$ is denoted: $\text{quality}_{\text{aesthetics}}(\ell)$

Label visibility. This aspect represents how well a label is visible on the map. It is influenced by other features and labels on the map. The visibility quality of a label ℓ is given by: $\text{quality}_{\text{label}-\text{vis}}(\ell, F, L)$

Feature visibility. This aspect represents how well a feature is visible on the map. It is influenced by other features and labels on the map. The visibility quality of a feature f is given by: $\text{quality}_{\text{feature}-\text{vis}}(f, F, L)$

Label-feature association. This aspect represents how clear it is that a particular feature and label are associated. It is partly influenced by other features and labels on the map. The association quality of a feature-label pair $\langle f, \ell \rangle$ is given by: $\text{quality}_{\text{association}}(\langle f, \ell \rangle, F, L)$

Note that the four categories are not only distinct conceptual (or perceptive) notions, the quality functions show that the parameters needed to evaluate each of them are different, too. This supports the claim that the four categories have only little overlap. Generally, the labels of L whose geometry attribute is set to UNLABELED will not influence visibility or association of other map objects. The quality function for label placement on the whole map is a combination of the qualities of the individual features and labels. The quality function has the form:

$$\begin{array}{rclll} \text{Quality}(P \cup F^{-}) & = & \gamma(& \bigoplus_{\ell \in L} & \text{quality}_{\text{aesthetics}} & (\ell), \\ & & & \bigoplus_{\ell \in L} & \text{quality}_{\text{label}-\text{vis}} & (\ell, F, L), \\ & & & \bigoplus_{f \in F} & \text{quality}_{\text{feature}-\text{vis}} & (f, F, L), \\ & & & \bigoplus_{\langle f, \ell \rangle \in P} & \text{quality}_{\text{association}} & (\langle f, \ell \rangle, F, L) \quad) \end{array}$$

The $\bigoplus$-combinators iterate over all elements of the corresponding set, and γ weighs the contribution of each quality category according to the given application. In technical maps, for instance, the visibility of labels and a good label-feature association could be more important than aesthetical issues or the visibility of objects that constitute the map background. Both $\bigoplus$ and γ could simply return the sum of their arguments.

4 Classification of names placement criteria

The general form of the quality function for label placement has been given, but we haven't yet defined the four partial quality functions. Before we can do this we shall summarize all criteria for label placement we found in the literature and classify them. We examined papers by Imhof (1975), Alinhac (1962), and Yoeli (1972). Our classification is according to whether and how the criteria can be incorporated into our model. A criterion can either be enforced automatically by our restrictions on the labeling possibilities, or it is represented in one of the four quality functions, or we have to ignore it in our model due to the additional complexity its integration would cause. Some of the criteria in the latter class will be discussed in Section 5.

4.1 Criteria handled by a-priori restrictions

1 The model does not consider fuzzy boundaries, so these boundaries must be made discrete.
2 Syllabification of names is not permitted. According to the model, names can never be broken into parts.
3 Names of features beyond the map border should be placed in the margin, names should not be placed in the fold between the pages of a book. However, our model assumes a rectangular map without margins and folds.
4 Spacing between letters should not be used for point or line labels.

4.2 Criteria handled by input specifications

1 Different font types and font sizes have to be used for the labels to reflect the classification and hierarchy of the objects and their graphical association on the map.
2 The type series should be easily grasped visually and be easily seen and read continuously. Do not abuse the choice in type styles.
3 Font sizes have to be chosen such that labels can be placed easily.
4 Apart from geometry, a classification in type for each feature should be made, depending on the scale of the map, for example. This is needed because on a small-scale map, a small area feature may have to be treated as a point, whereas on a large-scale map a river may have to be treated as an area. In this case the label should be placed inside the river.

4.3 Criteria that influence aesthetic quality

General

1 Labels should follow the reading direction. Intuitively, the label should be placed such that the reader does not have to tilt his or her head to read it. For example, letters should not be placed upside down.

Point label

1 Point labels should always be placed horizontally (respecting the coordinate system used for the map). This also holds for mountain peaks, it is considered bad practice to place the label in an arc below the peak.

Line label

1 Do not allow a line label to be complicated or extremely curved.
2 A line label should be as horizontal as possible.

Area label

1 A horizontal placing is preferred over a tilted placing.
2 Tilted area labels should be slightly curved.
3 No indecision in placing the label: if bent and crooked, then distinctly bent and crooked. The label should be clearly tilted, or not tilted at all (horizontal). Likewise, it should be clearly curved, or not curved at all.
4 The bending of the label should be simple: the radius of curvature of the area label should be constant for each word and it should not span an arc of more than 60 degrees.
5 If possible, bend curved labels somewhat towards the horizontal.
6 Only long area labels should be allowed to curve twice.
7 If no other placement is possible, an S-curve is allowed, but it should be composed of two circular arcs of the same radius.
8 The spacing between letters should preferably be less than twice (Imhof) or four times (Alinhac) the font size.

4.4 Criteria that influence label visibility

General

1 Labels should be placed either wholly on land or wholly in the water.
2 Overlap between a name and other features should be avoided.
3 If a label has to intersect another label, it is preferred that this happens at the boundary between units of compound words, see Figure 1.

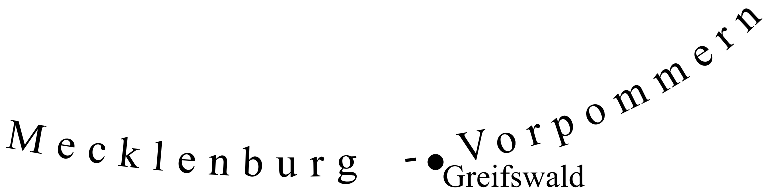

Figure 1: Place label at the boundary between units of a compound word.

Point label

1 Do not intersect point labels with a grid line.

Line label

1 Place line labels not too close to their feature.

Area label

1 Minimize overlap of area labels, including their spacing. This holds especially for labels of the same type face and size.
2 The amount of spacing can be decreased when the background has a low contrast with the label or when the background is dense.
3 Do not cross area labels with other labels at right angles.

4.5 Criteria that influence feature visibility

General

1 Names should disturb other map contents as little as possible. Avoid covering, overlapping, concealment. This holds especially for bifurcations, junctions, crossings and other important map features.

Line features

1 If a label of some feature must intersect a line feature of the same color, then avoid small angles between the label baseline and the line feature.

4.6 Criteria that influence label-feature association

General

1 Names should assist in revealing spatial location, territorial extent, connections, importance and differentiation of objects.
2 When a feature is located next to a line feature, the feature and its label should lie on the same side of the line feature.

Point label and feature

1 Place point label in order of preference, which is top right, bottom right, top left, bottom left, center top, center bottom.
2 A point label should be placed near its feature.
3 Names of coastal towns should be placed in the water.

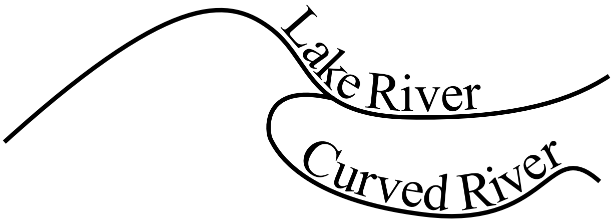

Figure 2: Line labels should follow the curvature of the line they label. They should be placed at junctions to indicate whether or not the line keeps its name after the junction.

Line label and feature

1 Place a line label such that the shape of the label follows the curvature of the line, see Figure 2.
2 Place line labels at the junction of rivers, since the name of the river may change there, see Figure 2.
3 Do not place a line label too far from its feature. The distance between the line feature and its label should be about half the font size.
4 Place line labels where there are the fewest objects.
5 Place line label over the line feature, not under it, see Figure 3.
6 Avoid labeling lines such that there are other features between the line and its label, see Figure 3.

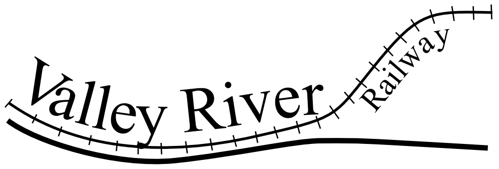

Figure 3: The label Valley River is badly placed because there is another feature between the label and the feature it labels. The label Railway is badly placed because it is below its feature.

Area label and feature

1 Extent and position of a label should indicate extent and form of an area. Bend and spread an area label such that it is spread across the shape of the area. The label should occupy approximately two thirds of the length of the line that extends the label across the area.
2 Expand a name towards the boundaries of the area, leaving a space of at least 1.5 times the font size at the ends.
3 A label is allowed to extend beyond the boundary of its area feature if it cannot easily be mistaken to be a label of another feature.

5 Possible Extensions

This section discusses a number of additional criteria for high quality labeling that cannot be incorporated easily into the classes of the previous section. We make a distinction in criteria that require more quality functions than the four given, and criteria that require an extension of the labeling model. Some of the criteria of the second type could be incorporated without much effort, while others are more difficult to evaluate.

Extensions that require more classes than the four considered

1 **Evaluation of regularities:** Label clusters and patterns made by labels should be avoided. For example, labels should not be evenly dispersed, and in dense areas labels should be centrifuged outwards. However, in order to keep the present model simple we chose not to evaluate several labels simultaneously, which is needed to detect clusters and patterns.

2 **Alterations to features:** When a label crosses a line feature and there is little contrast between the label and the feature, the line feature should be interrupted where the label crosses it. (However, if removing the background when a label is placed across a line feature is implied, this criterion would fit into the feature-visibility class and no alterations would be necessary.)

Extensions that require a more complicated model

1 **Multiple labels:** Sometimes it is useful to have multiple labels, for example an elevation and a name for a mountain peak or names in multiple languages for a city. Also, for line features, labels should be repeated at suitable intervals along the line feature. According to our model, each feature can only have a single label.

2 **Alternative label shapes:** In our model each name is placed on a single line in the label. Splitting names over multiple lines can be desirable if they consist of multiple words. Another case occurs when a point feature is on both sides of a line feature (for example a large city with a river running through it). Then the label should be split and placed on either side of the line feature.

3 **Separation of geometry and feature type:** Sometimes the label of an area or point feature should be placed as a line label. This is for example the case when the area label of a glacier has to indicate the direction of the flow of the ice. Another example is a mountain pass, which is labeled as a point but with the label giving some idea about the location of the pass. Note that this case is different from point 4 in Section 4.2 since there the classification could be done implicitly depending on the scale of the map and the fontsize used for labels.

4 **Polar coordinate systems:** The labeling of polar maps is a separate case, where labels have to be placed differently according to their distance and position from the pole. Our model does not consider this case.

5 **Different letter spacing:** Sometimes one can position a point label with a break to make room for some other feature (for example a contour line). This is not only considered bad practice, it is also not possible in our model where spacing is equal between all letters. It can be desirable however for a name which consists of multiple words to place the label with large spacing between the words and less spacing between the letters of a word.

6 **Fuzzy boundaries:** Our model does not consider fuzzy boundaries, so these boundaries must be made discrete.

7 **Explicit specifications of map dimensions, margins etc.:** Names of features beyond the map border should be placed in the margin, and names should not be placed in the fold between the pages of a book. However, our model assumes the map to be a rectangular area without margins.

8 **Relaxing restrictions of point-label shape:** In the model, point feature labels can only be rectangular. Sometimes more can be desired, for example if there are many labels nearby, it is best to place labels of coastal towns in the water and curve them away from horizontal.

9 **Special treatment of contour lines:** Contour lines can place extra demands on the placement of labels. For example, numbers indicating the height of the contour line should be directed such that they make up a “ladder” where the top of each label points in the direction of increasing height.

10 **Semantic aspects:** Different treatment of features with different semantics is not possible in our model. For example, administrative boundaries may be interrupted by labels, but rivers should not. This difference cannot be incorporated in the model.

6 An example function for name placement quality

This section describes a more detailed formalization of the quality function for label placement, that will supply a value indicating the quality of a label placement in the sense of Section 3.

First we decide to let the combinators $\bigoplus$ of Section 3.4 return the sum of their arguments. The γ-combinator adds up the four individual quality functions without weighing them, which corresponds to a map purpose where the four qualities are considered equally important. The general form of our quality function thus is:

$$\begin{array}{rrlll} \mathrm{Quality}(P \cup F^-) & = & \sum_{\ell \in L} & \mathrm{quality}_{\mathrm{aesthetics}} & (\ell) \\ & + & \sum_{\ell \in L} & \mathrm{quality}_{\mathrm{label-vis}} & (\ell, F, L) \\ & + & \sum_{f \in F} & \mathrm{quality}_{\mathrm{feature-vis}} & (f, F, L) \\ & + & \sum_{\langle f, \ell \rangle \in P} & \mathrm{quality}_{\mathrm{association}} & (\langle f, \ell \rangle, F, L) \end{array}$$

The summations are done over all labels that should be on the map, all features that are on the map, and all feature-label pairs that should be on the map, but where labels could have been omitted by the label placement method. Even though the quality function returns a value, and, because of it, allows to compare different labelings of the same map, the values should be interpreted on an ordinal scale. The values should suggest a reasonable ranking among the different labelings, but should not suggest how much better or worse some labeling is.

Next we describe how the four quality functions can be defined. For each quality function we give a simple realization and an extended version. The extended version takes into account more of the criteria listed in Section 4 than the simple version.

For simplicity we choose to let each quality function take values from 0 (lowest quality) to 100 (highest quality) for each feature or label. The priorities of the labels or features could weigh these quality values to incorporate the fact that certain features and labels are more important than others. We omit this issue from further consideration, since the adaptations necessary to incorporate priorities are straightforward. Recall that a label is considered to be a rectangular or curved shape, and that only for area labels spacing between letters of a word is allowed.

6.1 The aesthetic quality function

In our model point labels have a fixed shape, hence the aesthetic quality has the default value of 100 for point labels. For line and area labels it is determined by the shape. The aesthetic quality of a missing label is by definition equal to 100. The criteria of Imhof (1975) and Alinhac (1962) imply that a label should not have more than one inflection point and that the curvature of a label should be small.

Simple version. We define the aesthetical quality of a label ℓ as follows:

$$\text{quality}_{\text{aesthetics}}(\ell) = \begin{cases} 100 & \text{if the baseline has at most one inflection point and curvature at most the curvature of a circle with radius the height of the text;} \\ 0 & \text{if the baseline has more than one inflection point and curvature greater than the curvature of a circle with radius the height of the text;} \\ 50 & \text{otherwise.} \end{cases}$$

Extended version. We first show how *fading functions* help set up more realistic quantifications. We also argue why a quality function is given as a *multiplication* of the different criteria that are involved. These aspects also show up later in the other three extended quality functions, but here we treat them in the context of the extended quality function for aesthetics.

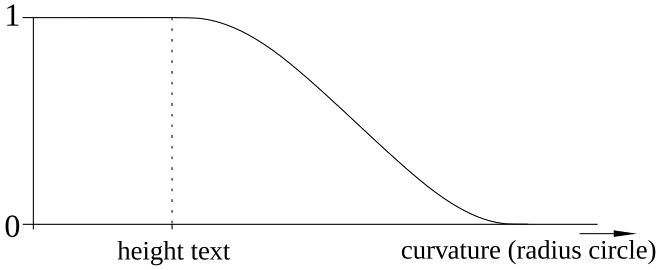

Figure 4: A fading function for curvature.

Fading functions. For our extended example we want to be able to smoothly differentiate quite good labelings from not so good labelings. For this purpose we use a *fading function*, which maps a continuously changing variable to a multiplication factor between 0 and 1 with which the quality value is multiplied, due to this variable. Figure 4 depicts an example of a fading function which maps the maximal curvature of a given label to a value that gives the fraction of the quality. From the figure it is clear that the best quality is obtained if the curvature is less than or equal to the curvature of a circle with radius the height of the text. A slightly higher curvature is not that bad, but the quality deteriorates when the curvature increases, because the multiplication factor becomes smaller.

For the aesthetics function we can consider the following factors which influence it:

- Number of inflection points.
- Average or maximal curvature.
- Average or maximal tilt.
- Maximal angular difference.[1]

Of these variables, the number of inflection points varies discretely, and the other three vary continuously.

[1] The maximal angular difference of a label is the maximal difference in angle from a point on the baseline and the starting angle of the baseline.

Multiplication. The different aspects of the quality function (such as tilt, curvature etc.) have to be combined to give the total quality value. Care must be taken in the way these aspects get combined. There are two natural ways for this: addition and multiplication. The addition formula has the form

$$\text{quality}_{\text{aesthetics}}(\ell) \quad = \quad w_1 \cdot \mathit{mapAspect}_1(\ell) \;+\; w_2 \cdot \mathit{mapAspect}_2(\ell) \;+\; \ldots$$

where the *mapAspect* functions return a value in the interval [0, 1], and the weights $w_1, w_2, \ldots$ should sum up to 100. The multiplication formula has the form

$$\text{quality}_{\text{aesthetics}}(\ell) \quad = \quad 100 \;\cdot\; \mathit{mapAspect}_1(\ell) \;\cdot\; \mathit{mapAspect}_2(\ell) \cdot\; \ldots$$

where the *mapAspect* functions again return a value in the interval [0, 1]. We chose to multiply the aspects. This has two advantages. Firstly, it makes the aspects to be independent. We assume the influence of the quality caused by for example the curvature of the label is independent from other factors, such as the tilt. The effect of having a bad or good curvature is captured by letting the total quality of the label continuously change between zero and the maximum quality that was achieved by all the other aspects. In the addition formula, any aspect can only influence the aesthetic quality partially, and a very bad aspect cannot make the aesthetic quality zero. The second advantage of multiplication of the aspects is that the additional weighing parameters $w_1, w_2, \ldots$ aren't needed.

In the multiplication method it is possible to make sure that some aspect cannot result in a quality value of zero. This is done by letting the corresponding *mapAspect* function return a value in a subinterval of [0, 1], for example the interval [0.7, 1] if the aspect shouldn't degrade the total quality to below 70% of what the other aspects give.

Returning to the extended example for the aesthetics quality function, each of the four aspects listed before can degrade the quality of the label. We set up a function for each of these aspects which maps the input to a value between zero and one and multiply the functions to obtain their combined quality measure. The (discrete) function *mapInf* that maps the number of inflection points to the quality fraction may look like: $\mathit{mapInf}(0) = 1$, $\mathit{mapInf}(1) = 1$, $\mathit{mapInf}(2) = 0.75$ and for $i > 2$ we let $\mathit{mapInf}(i) = 0.5$. Note that we choose to limit the effect of many inflection points to a degradation of the quality by 50 percent, since it would be undesirable to allow the quality to become zero if there are many inflection points but the label is good according to the other aesthetic aspects. The fading functions for tilt and maximal angular difference are similar to the fading function for curvature and can be designed in a manner that is appropriate for the kind of map at hand.

More concretely, the quality function for the extended example will look something like this:

$$\begin{array}{rcll} \text{quality}_{\text{aesthetics}}(\ell) & = & 100 & \cdot \;\; \mathit{mapInf}(\mathit{inflection}(\ell)) \\ & & & \cdot \;\; \mathit{mapCurv}(\mathit{curvature}(\ell)) \\ & & & \cdot \;\; \mathit{mapTilt}(\mathit{tilt}(\ell)) \\ & & & \cdot \;\; \mathit{mapAng}(\mathit{angDiff}(\ell)) \end{array}$$

where *mapInf* maps discrete values to a quality fraction, *mapCurv*, *mapTilt* and *mapAng* are fading functions, and *inflection*, *curvature*, *tilt* and *angDiff* return, respectively, the number of inflection points, the (average or maximal) curvature, the (average or maximal) tilt of the label, and the maximal angular difference. It is easy to extend this quality measure to incorporate more aspects by constructing the appropriate fading function. For example, in the case of area labels, spacing becomes important. If a label is spaced much too wide or much too dense, the quality of the label degrades. This can also be captured in a fading function.

Often it is especially important that the first part of the label is labeled well and it is less important whether the rest is labeled equally well. In that case the first part of the label can contribute more to the quality measure than the rest. If one wants to weigh the initial portion of the label more heavily, then this can easily be done by breaking the label in two at the desired point. Suppose ℓ is the initial label and we break it in two to obtain ℓ_1 and ℓ_2. We can now use our previous quality measure to obtain a weighted quality function, for example:

$$\text{quality}^{\text{weighted}}_{\text{aesthetics}}(\ell) \quad = \quad \frac{\text{quality}_{\text{aesthetics}}(\ell_1) + \text{quality}_{\text{aesthetics}}(\ell_2)}{2}$$

The division by 2 ensures that the value remains between 0 and 100.

6.2 The label visibility quality function

Text is considered a text block with a certain area. It can be disturbed if it overlaps with other text or if line features or area boundaries run through it. A line or boundary is considered to disturb all parts of the text block within a certain small distance from the line or boundary. So there is a buffer area around the line or boundary.

Simple version. For a label ℓ we define $\text{quality}_{\text{label-vis}}(\ell, F, L)$ as the percentage of the area of ℓ that is overlapped neither by other labels nor by the buffers of line features and boundaries, see Figure 5. If a label is missing, its visibility quality is defined to be 0.

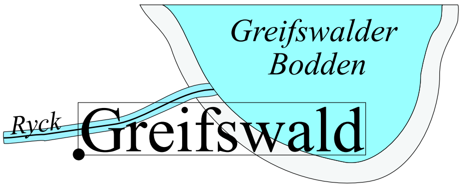

Figure 5: A simple label visibility quality function measures the percentage of the area of a label (here “Greifswald”) that is not covered by other labels or buffers around features.

Extended version. In the extended example for label visibility we can make a distinction between basic visibility of the label, distraction from the label, and undesirable interaction between text labels. Basic visibility includes contrast between text and background, and intersection with other text and features. Distraction includes a background which itself has highly contrasting parts. The undesirable interaction between two labels can occur when they have the same baseline and are close together.

It would carry too far to deal with color models, contrast, and perception issues here, so our extended example will be restricted in these aspects.

Average background color. We average all colors under the label to a single color, then use a function or table $contrast(\ell, c)$ to identify the contrast between the label color and the average background color c.

Intersection of labels and features with the same color. The function $intersect(\ell, F, L)$ returns the total area of displayed point, line, and area features under the label and of the same color, as a percentage of the total area of the label.

Distraction of background under the label. A measure for perceived density and complexity of the map objects under the label is needed. One could use, for instance, the local feature density by Mackaness (1995) or the clutter function by Jansen et al. (1998). It would carry too far to treat these in detail here. We assume that a local feature density function $LFD(\ell, F, L)$ is given.

Undesirable text interaction with other labels. This occurs only for labels with approximately the same color, font size, and font type. Interaction occurs when the end of one label seems to continue smoothly as the beginning of another label. The function $\mathit{interact}(\ell, L)$ is 1 if some other label has undesirable interaction with ℓ, otherwise it is 0.

For area label features we do not only have the label, but also the area in between the individual letters which may cause distraction from the label. It is most serious when the letters themselves are disturbed, it is moderately serious when the area in between is disturbed. For the area label visibility we can take this distinction easily into account.

The function *mapContrast* is a fading function that maps the value of *contrast* to a value in the interval [0, 1], where 0 represents no contrast and 1 represents a high contrast. Similarly, we use fading functions *mapIntersect* and *mapDistract* to convert the values from *intersect* and *LFD* to the interval [0, 1] or a subinterval of it. The discrete function *mapInteract* is used to map the output of *interact* to the same interval.

Using the multiplication principle, the extended quality function for label visibility is

$$\begin{aligned}\text{quality}_{\text{label-vis}}(\ell, F, L) \quad = \quad 100 \quad & \cdot \quad \mathit{mapContrast}(\mathit{contrast}(\ell, F, L)) \\ & \cdot \quad \mathit{mapIntersect}(\mathit{intersect}(\ell, F, L)) \\ & \cdot \quad \mathit{mapDistract}(\mathit{LFD}(\ell, F, L)) \\ & \cdot \quad \mathit{mapInteract}(\mathit{interact}(\ell, L)) \; .\end{aligned}$$

6.3 The feature visibility quality function

Simple version. Since point features are generally small, they are visible only if no label intersects them. So we define the visibility quality of a point feature to be 100 if it is not intersected by a label and it is 0 otherwise. We define the visibility quality of a line feature as the percentage of its length that is not covered by labels. Similarly, for area features the visibility quality is defined as the percentage of the area that is not covered by labels. Here we exclude the label of the area itself.

Extended version. There are only a few criteria listed in the literature concerning feature visibility as a result of label placement, see Section 4.5. For a feature, labels may disturb geometric, topological, and semantic properties. Here we concentrate on the geometric properties. From the input to the evaluation, it is for instance impossible to find out how important the topological interrelation between two features really is, and therefore also how badly a label could disturb an interrelationship. If a forest borders a lake, then both lake and forest have their own importance, but one cannot tell how important the fact is that the forest borders the lake. As a rule of thumb, however, we can say that junctions of linear features like roads and rivers have a special importance. If we assume that such junctions are separate point features of the feature class with their own importance, then no special provisions need to be made. The most important topological relationships will then be taken into consideration in the evaluation of label placement quality.

The extended model need not be different from the simple model. Point features have visibility quality 100 if they are visible and 0 if they are obscured. Line and area features have visibility quality equal to the percentage that is not obscured.

For point and line objects, we could add a distraction component that takes distraction by features and labels in the neighborhood into account. The quantity of distraction decreases with the distance to the feature.

6.4 The association quality function

The association quality of a feature f and its label ℓ is good if the feature and its label are close, but no other feature is close to the label ℓ, and no other label is close to the feature f. We use this general observation both in the simple and the extended quality function for association.

Simple version. For a point p and its label ℓ, we define:

$$\text{quality}_{\text{association}}(\langle p,\ell\rangle, F, L) = \begin{cases} 100 & \text{if } p \text{ is within half the text height from } \ell\text{, no other} \\ & \text{point is within once the text height from } \ell\text{, and no} \\ & \text{other label is within once the text height from } p \\ 0 & \text{otherwise} \end{cases}$$

For line features the situation is more complicated because we would like the label to be close to the line over its full length. Therefore, we define the quality by thickening the line with a buffer of width $1\frac{1}{2}$ times the text height, and take the percentage of the label area that lies within this buffer.

For area features the situation is yet more involved because the label should be close to most of the area. Intuitively, albeit somewhat simplistic, the part of the area that is associated to the label lies within a certain distance from it. Again the idea is to use a buffer, but this time we take a buffer around the text. Its width is chosen twice the text height. The quality of association of an area feature a and its label ℓ is defined as:

$$\text{quality}_{\text{association}}(\langle a,\ell\rangle, F, L) \quad = \quad 100 \cdot \frac{\text{Area}(\text{buffer}(\ell) \cap a)}{\text{Area}(\text{buffer}(\ell))}\ .$$

If the label is missing, then the association quality is defined to be 0. Note that this definition also makes sense when ℓ or buffer(ℓ) is larger than the area a.

Extended version. For the association of a label with its feature we use a two-level approach: we devise a function *plainAssoc*(f,ℓ) with range $[0,1]$ that takes any feature and any label, and returns how well they associate without taking into account the other labels or features. Then we use *plainAssoc* to define association on the map: association of a label and feature in the map context is good if *plainAssoc* returns a high value for f and ℓ, it returns a low value when applied to ℓ and any other feature than f, and it returns a low value when applied to f and any other label than ℓ.

The association can be described by the following function that takes into account the three components that influence association:

$$\begin{aligned} \text{quality}_{\text{association}}(\langle f,\ell\rangle, F, L) \quad = \quad 100 \quad & \cdot \quad \mathit{plainAssoc}(f,\ell) \\ & \cdot \quad (1 - \max_{\hat{f}\in F, \hat{f}\neq f} \mathit{plainAssoc}(\hat{f},\ell)) \\ & \cdot \quad (1 - \max_{\hat{\ell}\in L, \hat{\ell}\neq \ell} \mathit{plainAssoc}(f,\hat{\ell}))\ . \end{aligned}$$

The value of the function *plainAssoc*(f,ℓ) depends on several aspects, in the first place the distance between f and ℓ. Other aspects are color, feature type and geometry, and label type and geometry. When f is a point feature and ℓ is a curved area label with spacing, then f and ℓ will never be associated with each other, regardless of distance and color. Similarly, no red label will be associated to a blue river. First we define the function *plainAssoc* for point features, then we will show how this function needs to be modified for line and area features.

For a point feature f and any label ℓ (point, line, or area label), we let

$$\begin{aligned} \mathit{plainAssoc}(f,\ell) \quad = \quad & \quad \mathit{mapDist}(\mathit{dist}(f,\ell)) \\ & \cdot \quad \mathit{mapContrast}(\mathit{contrast}(f,\ell)) \\ & \cdot \quad \mathit{straight}(\ell) \\ & \cdot \quad \mathit{sizeDiscrepancy}(f,\ell) \\ & \cdot \quad \mathit{preference}(f,\ell). \end{aligned}$$

Here, *dist*(f,ℓ) is the smallest distance between f and ℓ, and *mapDist* is a fading function that converts the distance to the interval [0, 1]. The difference in color is captured by the function *mapContrast*(*contrast*(f,ℓ)), the discrepancy in size of the point feature and the text height by *sizeDiscrepancy*(f,ℓ), and *straight*(ℓ) is a function that returns 1 if ℓ is straight and 0.5, for example, when ℓ is curved. Finally, the function *preference* uses the angle from the point feature to the label to incorporate preference positions. The angle is converted to the interval [0.8, 1], for example.

For a line feature and any label, we let *plainAssoc* be defined in almost the same way, but *dist* should be interpreted as the directed one-sided Hausdorff distance from the label ℓ to the line feature f, i.e. $\max_{p\in\ell}\min_{q\in f} d(p,q)$, where $d(p,q)$ is the Euclidean distance of two points p and q in the plane, see Huttenlocher et al. (1993). Furthermore, we omit the factor $\mathit{straight}(\ell)$, and we include a factor for shape discrepancy, which can be based on the standard deviation of the distance from the baseline of the label to the line feature f.

For an areal feature f and any label ℓ the function *plainAssoc* can be defined in a similar manner. The distance $\mathit{dist}(f,\ell)$ is defined as the average distance of the points on the boundary of the label to the boundary of the area. This definition would give area labels that are in the middle and have the same shape as the area a higher value than labels that aren't.

7 Conclusions

In this paper we discussed the evaluation of how well a label placement method performs its task on a given input. This involved developing a general model, a formalization, of the shapes of label types, and of the input and output of a label placement method. The evaluation is given as a function that maps the input and output of a label placement method to a value representing the quality. The main use of such a function is to compare how well different label placement algorithms perform their task on the same input.

We listed roughly sixty criteria for good label placement and classified them according to aesthetic quality, label visibility, feature visibility, and label-feature association. Then we gave a concrete example of a quality function, fully specifying the quality of a label placement method on a given input. We kept this function simple; our goal was to capture the most important criteria in a small number of geometric concepts that can be computed automatically. These concepts include inflection points of curves, curvature, distance, buffers (dilation), and area of overlap. We also presented an example of a more extended quality function that dealt with more of the labeling criteria, and that should incorporate these criteria better than the simple function.

Perhaps the most important next step towards high-quality label placement is testing how well quality is represented by the quality function given. This requires the implementation of the functions in this paper, testing it on labeled maps, and cartographic expert judgement of the results. Since our quality function is designed to be transparent, possible shortcomings of the quality evaluation can be traced to the contributing components of the quality function, and adjustments can be made.

In the PhD thesis of Tycho Strijk [22], Chapter 9 contains the special case of street labeling using a quality function based on this paper. Figure 6 shows an example where several quality criteria have been taken into account, and labeling is done by simulated annealing using the quality function. It is clear that the labeling can be improved in some aspects, which can be done by incorporating more criteria in the quality function. A possibility excluded in our paper, but common in street labeling, is the stacking of labels. In the figure, one can see that stacking is allowed, but is avoided if possible in the case study of Strijk.

We believe that a formalization, a quantification, of all criteria for high-quality label placement will lead to a better understanding of the final goals for automated, high-quality label placement. Eventually this should lead to efficient, high-quality label placement methods themselves. This paper is a first step towards such a formalization.

Acknowledgements. Steven van Dijk, Tycho Strijk and Marc van Kreveld are supported by the Dutch Organization for Scientific Research (NWO). During this research Alexander Wolff was supported by the Deutsche Forschungsgemeinschaft (DFG) under grant Wa 1066/3-1. Furthermore, this research is supported in part by the ESPRIT IV LTR Project No. 28155 (GALIA).

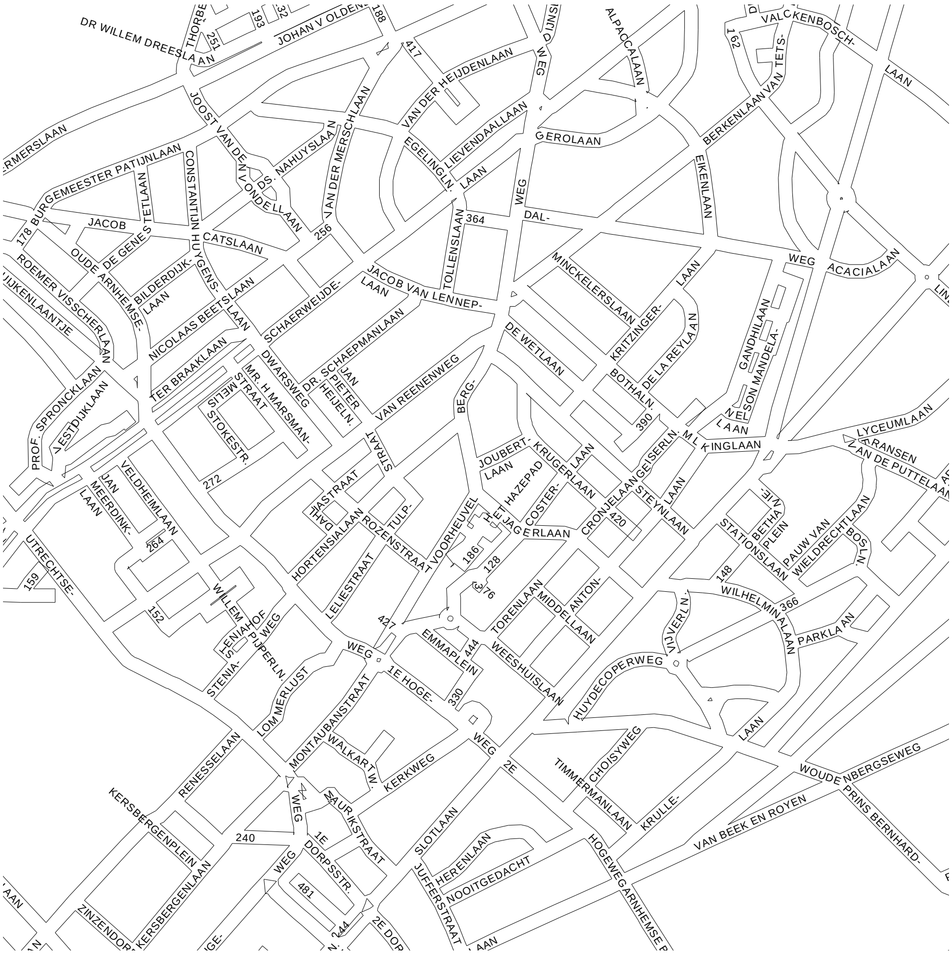

Figure 6: Street labeling based on quality functions and simulated annealing.

References

[1] John Ahn and Herbert Freeman. AUTONAP—an expert system for automatic map name placement. In *Proceedings International Symposium on Spatial Data Handling*, pages 544–569, 1984.

[2] Georges Alinhac. *Cartographie Théorique et Technique*, chapter IV. Institut Géographique National, Paris, 1962.

[3] Mathieu Barrault. An automated system for name placement which complies with cartographic quality criteria: The hydrographic network. In *Proceedings of the Conference on Spatial Information Theory (COSIT'97)*, volume 1329 of *Lecture Notes in Computer Science*, pages 499–500, Pittsburgh, PA, 1997. Springer-Verlag.

[4] Mathieu Barrault. *Le placement cartographique des écritures: résolution d'un problème à forte combinatoire et présentant un grand nombre de contraintes variées.* PhD thesis, Marne-la-Vallée University, November 1998.

[5] Anthony C. Cook and Christopher B. Jones. A Prolog interface to a cartographic database for name placement. In *Proceedings 4th International Symposium on Spatial Data Handling*, pages 701–710, 1990.

[6] Borden D. Dent. *Cartography*, chapter 14. Wm. C. Brown Publishers, 1996.

[7] Jeffrey S. Doerschler and Herbert Freeman. A rule-based system for dense-map name placement. *Communications of the ACM*, 35:68–79, 1992.

[8] Shawn Edmondson, Jon Christensen, Joe Marks, and Stuart Shieber. A general cartographic labeling algorithm. *Cartographica*, 33(4):13–23, 1997.

[9] ESRI. MAPLEX—a fully automated cartographic name-placement software. `http://www.esri.com/software/maplex/`, 1998.

[10] Evermap. Evername—an advanced cartographic label-placement software for MapInfo Professional. `http://www.evermap.com/evername.htm`, 1999.

[11] Stephen A. Hirsch. An algorithm for automatic name placement around point data. *The American Cartographer*, 9(1):5–17, 1982.

[12] Daniel P. Huttenlocher, Gregory Klanderman, and William J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:850–863, 1993.

[13] Eduard Imhof. Positioning names on maps. *The American Cartographer*, 2(2):128–144, 1975.

[14] Michiel Jansen and Marc van Kreveld. Evaluating the consistency of cartographic generalization. In T.K. Poiker and N. Chrisman, editors, *Proc. 8th Int. Symp. on Spatial Data Handling*, pages 668–678, 1998.

[15] Christopher Jones. Cartographic name placement with Prolog. *IEEE Computer Graphics & Applications*, 9(5):36–47, 1989.

[16] Gail E. Langran and Thomas K. Poiker. Integration of name selection and name placement. In *Proc. Auto-Carto 8*, pages 50–64, 1986.

[17] William A. Mackaness. A constraint based approach to human computer interaction in automated cartography. In *Proc. of the 17th Int. Cartographic Conference*, pages 1423–1432, 1995.

[18] MapText. Label-EZ for automated cartographic text placement—a high-performance productivity tool for the mapping industry. `http://www.maptext.com`, 1999.

[19] Itzhak Pinto and Herbert Freeman. The feedback approach to cartographic areal text placement. In P. Perner, P. Wang, and A. Rosenfeld, editors, *Advances in Structural and Syntactical Pattern Recognition*, pages 341–350. Springer-Verlag, New York, 1996.

[20] Mike Preuß. Solving map labeling problems by means of evolution strategies. Master's thesis, Fachbereich Informatik, Universität Dortmund, February 1998.

[21] Arthur H. Robinson, Joel L. Morrison, Phillip C. Muehrcke, A. Jon Kimerling, and Stephen C. Guptill. *Elements of Cartography*, chapter 22. John Wiley & Sons, Inc., 1995.

[22] T. Strijk. *Geometric Algorithms for Cartographic Lable Placement.* PhD thesis, Institute of Information and Computing Sciences, Utrecht University, 2001.

[23] Alexander Wolff, Lars Knipping, Marc van Kreveld, Tycho Strijk, and Pankaj K. Agarwal. A simple and efficient algorithm for high-quality line labeling. In David Martin and Fulong Wu, editors, *Proc. GIS Research UK 7th Annual Conference (GISRUK'99)*, pages 146–150, Southampton, 14–16 April 1999. Department of Geography, University of Southampton.

[24] Alexander Wolff and Tycho Strijk. A map labeling bibliography. `http://www.math-inf.uni-greifswald.de/map-labeling/bibliography/`, 1996.

[25] Pinhas Yoeli. The logic of automated map lettering. *The Cartographic Journal*, 9:99–108, 1972.