Towards an Evaluation of Quality for Label Placement Methods

Steven van Dijk
Dept. Computer Science
Utrecht University
steven@cs.uu.nl

Marc van Kreveld
Dept. Computer Science
Utrecht University
marc@cs.uu.nl

Tycho Strijk
Dept. Computer Science
Utrecht University
tycho@cs.uu.nl

Alexander Wolff
Institut für Informatik
Freie Universität Berlin
awolff@inf.fu-berlin.de

Abstract

The cartographic labeling problem is the problem of placing text on a map. It is composed of two phases. In the first, the question which map features should in principle receive a label is settled and the style (i.e. color and font) of these labels is determined. The second phase consists of the actual label placement. In this phase for each feature one has to decide whether there is in fact sufficient space and, if yes, the best location and shape of the label must be determined. This paper proposes a quality measure for the result of the second phase, allowing the comparison of label placement programs.

1 Introduction

For a human cartographer label placement is a tedious and 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. 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 give a classification for most requirements 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 to understand what contributes to good label placement in general. Secondly, it helps 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 as follows. In the next section we will discuss globally criteria for high-quality map labeling as described in the cartographic literature and list those that are 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. In Section 4 we give an example of a fully operationalized quality function that fits into our framework. The full version of this paper gives a more complete description and several possible extensions as well.

2 Aspects of lettering maps

A map—on paper or on a screen—consists of a number of geographic objects together with annotation. Geographic objects are objects that have coordinates associated to them, and can be zero-, one-, or two-dimensional. They can also be symbols, or composite objects. Annotation consists of labels, i.e. text associated to geographic 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, Alinhac, and Yoeli have each listed a number of requirements for high-quality labeling (Imhof, 1975; Alinhac, 1962; Yoeli, 1972). These requirements have been summarized in various textbooks and surveys, e.g. (Dent 1996; Robinson et al., 1995). Here we briefly list the major high-level rules again. Notice that not all aspects of these rules are relevant to the placement of text, for instance the choice of font.

- Legibility: Influenced by font size, font color (contrast), overlap with other labels and features, and label position relative to its feature. In addition, labels of different features should not be placed close to each other on a horizontal line.

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

- Harmony: 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. Furthermore, a particular variant and color should be chosen for all map features of the same type, for example, all rivers should have blue labels.

- Unambiguity: Involves avoiding text close to objects it does not correspond to, avoiding objects between a label and its object. 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 is 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 the position, orientation, shape, and extent: 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 or ocean. Other examples are map features with indeterminate boundaries, features whose boundary is not explicitly shown, and composite features like groups of islands.

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.

Yoeli (1972) was the first to incorporate position preferences in his algorithm. Later many others followed this. Hirsch (1982) used buffer circles around points. By forbidding labels to intersect these buffers, he prevented ambiguity. Ahn and Freeman developed a name placement system called AUTONAP that includes area labeling (1984). 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 not to disturb 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. Preferably labels are placed in positions with low 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 the river or boundary. 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 also 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. In the software package Maplex (1998), for each feature class one can set feature and label weights that determine which labels may hide which features. In addition, a user can set the size of label-to-feature and label-to-label buffers that determine the minimum separation between a label and other features or other labels.

Preuß (1998) proposed a fitness 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 don't 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 fitness function of Preuß also takes into account the preferential positions for a label around a point feature. 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.

2.3 Modeling aspects for the quality function

The quality function for label placement that we will develop is a special type of model. The concept 'model' is a very broad one, so we will specify first what characteristics and type of model we set out to define. Usually, a model is a simplified version of the real world, one that only captures the aspects that are essential to the use of the model. In our case, the 'real world' is the map including the labels, and the model is a function that represents the quality of the placement.

The output of the quality function should be easy to interpret. The easiest type of output is a *value* for the quality. This value should be the higher the better the labels are placed. 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 at least for all cases where one labeling is clearly better than another. If two human cartographers decide differently on which label placement is better, then the ordering of the labelings by the quality function can follow only one judgement, of course.

3 A specification for name placement quality

Our model aims for the properties simplicity, relevance, tractability, and understandability. *Simplicity* is a general asset of a model. The type of *relevance* we expect of the model 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 that 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 on only a few parameters whose tuning has predictable effects on the quality function output.

3.1 Elements of the map

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

- Point features, which can be point symbols or other 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 such as color, pattern, and width for lines. Note that there are many types of lines. 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 close to the features they are associated with.

All of these objects have an explicit position on the final map, and that map can be described purely by geometry (coordinates) and display style of its objects.

3.2 A-priori restrictions of label placement

Label placement is considered to be done by some method or algorithm. The a-priori labeling restrictions specify beforehand what type and shape of labels could be generated by the method. 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 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 limitation we make is that any feature receives at most one label on the map. Secondly, 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 won't treat marginal annotation. The other limitations deal with the allowed shape of labels, and are different for point, linear, and areal feature labels, so we treat these separately.

Point labels are text blocks that must align with parallels of latitude on the map. On large-scale maps, the parallels are (nearly) horizontal, and all point labels should be horizontally aligned text. The text itself, the type, typeface, font, and color of the text is fixed for each separate point feature. The text block is treated as a rectangular block, somewhat like a bounding box. If the parallels are curved, then the rectangle is also curved. The text block is completely specified by the coordinates of one corner at the start and bottom of the text, and the height and the width of the text.

Other than point labels, the text of line and area labels may follow curves. Each area label additionally allows specifying an inter-character distance.

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 to use, and which map projection to use. Intuitively, the map background has been drawn but no labels have been placed yet. However, which features should receive labels and in which font and size has been decided. Let $F = F^+ \cup F^-$ be the set of features, F^+ (F^-) the set of features that are (not) to be labeled, and L the set of labels. Note that features in F^- must also be part of the input since they must be considered when placing labels of other features.

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

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

The priority of a label may be different from the priority of its feature, because sometimes the visibility of a feature is more important than its label. The fourth 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 $< f,l >$ in P is specified if l could be placed. Note that the output contains more information than the labeled map itself, because from a map one cannot determine the priority of its features and labels, nor which labels were intended but could not be placed. This information is necessary to evaluate the quality of a label placement.

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 high-priority labels are involved. The input to our quality function consists of the output of the label placement method.

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 little overlap in criteria.

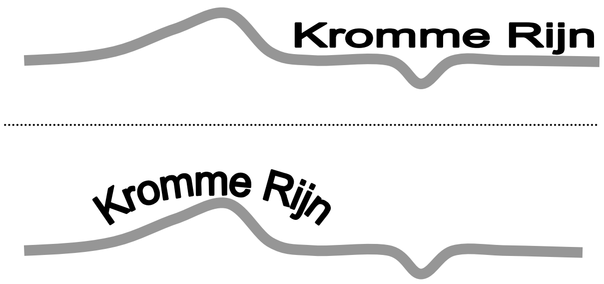

Figure 1. Aesthetics: Horizontal river labels are preferred.

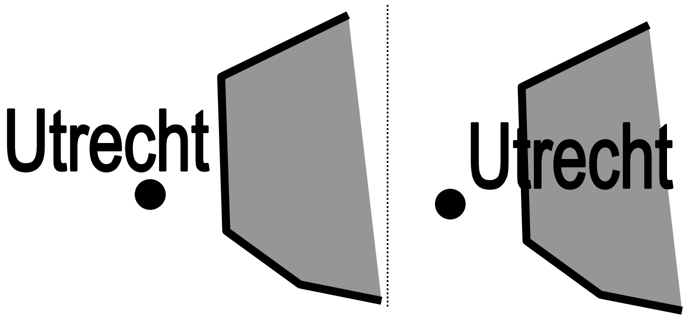

Figure 2. Label visibility: Background should not make label text unreadable.

Let $L^* \subseteq L$ be the set of all labels from L that were placed by the method. 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/or 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 position on the map, or by other map features. The aesthetic quality of a label $l \in L$ is denoted:
$\mathbf{quality}_{\mathbf{aesthetics}}(\boldsymbol{l})$

Label visibility.

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

Feature visibility.

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

Label-feature association.

This aspect represents how clear it is that a particular feature and label are associated. It is partly influenced by other map features and labels. The association quality of a feature-label pair $< f, l >$ is given by:
$\mathbf{quality}_{\mathbf{association}}(\, < \boldsymbol{f}, \boldsymbol{l} > , \boldsymbol{F}, \boldsymbol{L}^*)$

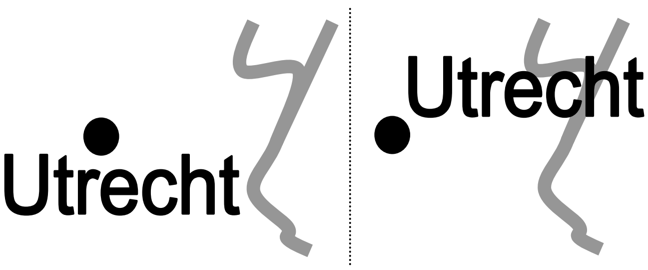

Figure 3. Feature visibility: labels should disturb other map contents as little as possible. Text should not cover important information like river junctions.

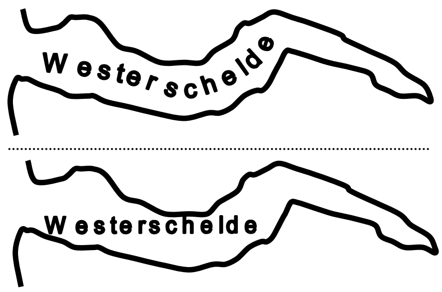

Figure 4. Association: Label shapes should assist in revealing spatial location.

Note that the four categories are not only distinct conceptual (or perceptive) notions, the quality functions indicate that the parameters needed to evaluate each of them are different too. This supports the claim that the four categories only have little overlap. The quality function for label placement of the whole map is a combination of the qualities of the individual features and labels. The quality function has the form:

$$\begin{aligned}\text{Quality}(P \cup F^{-}) \;=\; \gamma\,(\; & \oplus_{l \in L}\, \text{quality}_{\text{aesthetics}}(l),\\ & \oplus_{l \in L}\, \text{quality}_{\text{label-vis}}(l, F, L^{*}),\\ & \oplus_{f \in F}\, \text{quality}_{\text{feature-vis}}(f, F, L^{*}),\\ & \oplus_{<f,l> \in P}\, \text{quality}_{\text{association}}(<f, l>, F, L^{*})\;)\end{aligned}$$

The ⊕-combinators (e.g. summations) 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 is more important than aesthetics or the visibility of objects that constitute the map background.

4 An example function for name placement quality

This section describes a relatively simple quality function as an example. There are several ways to extend and improve upon the function to be outlined.

Firstly, we let the combinators ⊕ return the sum of their arguments. The γ-function adds up the four individual quality function values, corresponding to a map purpose where the four qualities are considered equally important. The form of the quality function is:

$$\begin{aligned}\text{Quality}(P \cup F^{-}) \;=\; & \textstyle\sum_{l \in L}\, \text{quality}_{\text{aesthetics}}(l)\\ & + \textstyle\sum_{l \in L}\, \text{quality}_{\text{label-vis}}(l, F, L^{*})\\ & + \textstyle\sum_{f \in F}\, \text{quality}_{\text{feature-vis}}(f, F, L^{*})\\ & + \textstyle\sum_{<f,l> \in P}\, \text{quality}_{\text{association}}(<f, l>, F, L^{*})\end{aligned}$$

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 the label could have been omitted.

Next we describe how the four quality functions can be defined. 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 and 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 here. In the four quality functions, the label is considered a text block that has some area.

4.1 The aesthetic quality function

With the labeling model and requirements in mind, aesthetic quality of point labels is not interesting, and for line and area labels it is determined by the shape. According to Imhof (1975) and Alinhac (1962) a label should not have more than one inflection point and the curvature of a label should be small (Figure 5). Therefore, we define the aesthetic quality of a label *l* as follows:

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

We define the aesthetic quality of a missing label to be 0, but another choice like 100 can actually be better in some cases.

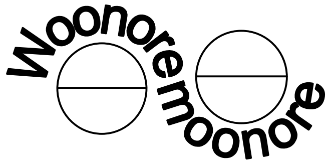

Figure 5. The curvature of a label should not exceed the curvature of a circle with radius equal to the text height.

Figure 6. A line feature disturbs the visibility of a label within some distance of the line.

Figure 7. The label isn't placed optimally because its buffer is partly outside the area.

4.2 The label visibility quality function

Text is considered as a text block with a certain area. This area 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, say, 1 millimeter, from the line or boundary (Figure 6). The region within some distance from an object is called the buffer or dilation of that object.

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

4.3 The feature visibility quality function

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. For line features we are interested in how much of their length is covered by labels. 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.

4.4 The association quality function

The association quality of a point feature and its label is good if the point *p* and label *l* are close, but no other point is close to the label *l* and no other label is close to the point *p*. We define:

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

For line features the situation is more complicated because we prefer labels that are close to the line over the full label length. Therefore, we define the quality by thickening the line with a buffer of width $1^1/_2$ 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, 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 (Figure 7):

$$\text{quality}_{\text{association}}(\langle l, a \rangle, l, F, L^*) = 100 \cdot \text{Area}(\text{buffer}(l) \cap a)) / \text{Area}(\text{buffer}(l)).$$

If the label is missing, then the association quality is defined to be 0.

Conclusions

This paper discussed the automated evaluation of how well a label placement method performed its task on a given input.

This involved developing a general model, a formalization, of the shapes of types of labels, 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 for comparing how well different label placement algorithms perform their task on the same input.

In the full paper we list roughly sixty criteria for good label placement and classify them as aesthetic quality, label visibility, feature visibility, and label-feature association. Here we only gave some examples of these criteria for the four classes. Then we gave a concrete example of a quality function, specifying fully the quality of a label placement method on an input. We kept this function very 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.

Several types of extension to this research are possible. The evaluation function can be extended to incorporate more criteria, like taking color of features and text into account, and testing for regularity in label positions. The a-priori restrictions on labeling can be relaxed, for example by allowing more than one line per label. Furthermore, it is possible to generalize label placement evaluation to typography evaluation by including, for instance, the choice of font. Such extensions and the issues involved are discussed in the full paper.

We believe that a formalization, a quantification, of all criteria for label placement by Imhof and Alinhac 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 algorithms 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). Alexander Wolff is 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).

References

Ahn, J., and H. Freeman (1984). AUTONAP—an expert system for automatic map name placement. In *Proceedings International Symposium on Spatial Data Handling*, pages 544-569.

Alinhac, G. (1962). *Cartographie Théorique et Technique*, chapter IV. Institut Géographique National, Paris.

Cook, A.C., and Jones, C.B. (1990). A Prolog interface to a cartographic database for name placement. In *Proceedings Fourth International Symposium on Spatial Data Handling*, pages 701-710.

Dent, B.D. (1996). *Cartography*, chapter 14. Wm. C. Brown Publishers.

Doerschler, J.S., and H. Freeman (1992). A rule-based system for dense-map name placement. *Communications of the ACM*, 35:68-79.

Edmondson, S., J. Christensen, J. Marks, and S. Shieber (1997). A general cartographic labeling algorithm. *Cartographica*, 33(4):13-23.

ESRI (1998). *Maplex—Automatic Cartographic Name Placement Software*.

Hirsch, S.A. (1982). An algorithm for automatic name placement around point data. *The American Cartographer*, 9(1):5-17.

Imhof, E. (1975). Positioning names on maps. *The American Cartographer*, 2(2):128-144.

Jones, C. (1989). Cartographic name placement with Prolog. *IEEE Computer Graphics & Applications*, 9(5):36-47.

Langran, G.E. and T.K. Poiker (1986). Integration of name selection and name placement. In *Proc. Auto-Carto 8*, pages 50-64.

Pinto, I. and H. Freeman (1996). The feedback approach to cartographic areal text placement. In A. Rosenfeld P. Perner, P. Wang, editors, *Advances in Structural and Syntactical Pattern Recognition*, pages 341-350. Springer, New York.

Preuß, M (1998). Solving map labeling problems by means of evolution strategies. Master's thesis, Fachbereich Informatik, Universität Dortmund.

Robinson, A.H., J.L. Morrison, P.C. Muehrcke, A.J. Kimerling, and S.C. Guptill (1995). *Elements of Cartography*, chapter 22. John Wiley & Sons, Inc.

Yoeli, P. (1972). The logic of automated map lettering. *The Cartographic Journal*, 9:99-108.