IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITP4939

Intelligent Icon Positioning for Interactive Map-Based Information Systems

Georg Fuchs
University of Rostock, Institute for Computer Graphics, georg.fuchs@informatik.uni-rostock.de

Heidrun Schumann
University of Rostock, Institute for Computer Graphics, schumann@informatik.uni-rostock.de

ABSTRACT

The combination of map displays with icon techniques is well-suited for visualizing geo-spatial dependent multivariate data. The requirements for icon placement on the map, although not trivial, are not properly attended by most interactive visualization systems. In cartography, a number of methods for automatic text label placement and map generalization has been developed. These are focused on high-quality results, leading to long response times not suited for interactive environments. In this paper an interactive, scalable approach borrowing from cartographic methods is presented which is based on an iterative displacement of icons combined with a focus & context technique.

MOTIVATION

The use of graphical means for analysis and evaluation of large datasets is by now widespread and commonly accepted. Much of the acquired data includes some form of geographical reference or location coordinates. For the visualization of this spatial context, map presentations are a natural choice.

Very often more than one parameter value is recorded per measurement. Icon techniques are especially suited for the visualization of this so-called multivariate data. An icon is a graphical primitive that encodes the different parameter values in visual variables such as size, shape or color. Furthermore, this technique allows the exact placement of individual icons on the display area. It is therefore well-suited for integration into map presentations, where the map communicates the spatial-geographic reference of the abstract information encoded in the icon. While many publications in the field of visualization address problems pertaining the design of icons or the encoding of values with icons [2,14], nearly no papers consider the problem how to place icons on the map.

Generally, maps are structured hierarchically according to geographic and administrative actualities (e.g. countries, districts, ZIP code areas). These areas vary strongly in both shape and size, making the task of positioning icons a non-trivial one. Especially if information are present for densely clustered map areas, this can quickly result in an overloaded representation and overlaps of icons, causing loss of information (Figure 1). To avoid this information loss, the following problems have to be solved:

- the mutual overlap of icons,
- the occlusion of information on the map background, and
- ambiguities in the association of individual icons with the correct geographical coordinates or area.

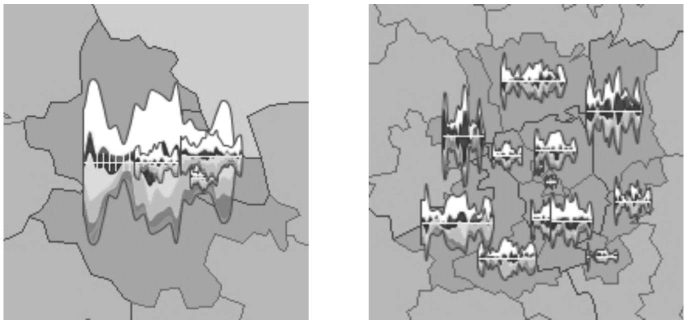
In current interactive map-based information systems, these aspects are often addressed insufficiently, or not at all. In the majority of cases, the position of icons is determined by simple rules, e.g. placing the icon within the bounding box of a map area. Figure 1 shows health data as ThemeRivers [7] on a map. The width of the 'river' indicates the total amount of cases over time, with individual color bands comprising the river showing the number of cases for specific diseases. Here, the icons have been positioned in the center of the bounding box (minimal enclosing rectangle) of the respective area borderline polygons. The icon size therefore depends on the size and shape of the associated map area. In this figure, it is difficult to compare case occurrences between different areas because of overlapping and different icon sizes. This shows that a poor positioning leads to non-expressive representations that are difficult to interpret.

A closely related problem in cartography is the task of symbol and name (label) placement on maps. All labels on a map should be clearly readable and there must be no ambiguities as to which map feature they are adorning. Since the advent of GIS numerous algorithms for automatic text label placement have been developed, from rule-based expert systems [1, 4], physical models [8] and methods of linear programming [10, 15] to stochastic methods such as Simulated Annealing [5] and genetic algorithms [12]. The majority of these algorithms concentrate on labeling of point features rather than on labeling of areas as we need.

In map labeling, it is common not to label all objects if some names could not be placed without overlaps. The information loss of the missing label is less than the loss that would be caused by the overlap. Contrary to this, in information visualization the completeness of a presentation is stipulated, i.e. all selected information records' icons must be presented and interpretable in the picture. Therefore, the complexity of a (complete) solution is therefore even higher compared to map labeling.

Moreover, paramount goal of the mentioned methods are high-quality results as required for static, i.e. non-interactive presentations, preferably coming close to that of manual processing by man. Optimization of system response times is not a key issue. In our context, response times are an important factor. Here, small deficiencies in icon placement can be dynamically corrected based on user interaction.

Figure 1: Result of positioning the icons to fit the bounding box of their associated map area. Some icons are partially or completely occluded (left), while different icon sizes make comparisons difficult (right).



However the avoidance of overlaps, and thus information loss, is the single most important goal in both cases.

One common method in cartography to avoid overlapping is displacement. Here too, a number of methods has been devised [9, 11]. An interesting approach is the spring force model [3].

Icon positions can be solved as well using a simplified but performant, labeling and displacement algorithm. The remaining spatial conflicts are then interactively solved by means of a suitable focus & context technique.

The paper describes this approach and is structured as follows. First in section 2 some basic terms are introduced, and a brief summary of the key properties of icons and their differences to map labels is given. Section 3 introduces an iterative positioning algorithm based on object displacement for interactive visualization systems and describes an implemented prototype. Section 4 concludes with a summary.

BASICS

Owing to space limitations, in this paper we focus on icons that are positioned with respect to map areas.

For placing an icon on a map area, two parameters are of interest: its position and the amount of display it occupies in the presentation. The icon position is given by a *reference point* in the map coordinate system. An icon is positioned on the map by translating its local center point into its reference point on the map. In order to ensure an icon is always associated with the correct map area, the reference point must always lie within the polygon defining that area's boundaries.

To assess the amount of display space an icon consumes on the map, the *contour* is used which describes the boundary of the icon's area. It is not necessary that it accurately matches the form of the icon. For positioning on maps an enclosing (adequately minimal) approximation is sufficient. The bounding circle (minimum-radius circle enclosing the entire representation) allows efficient detection of overlaps and is therefore used here.

One question arising is how to choose the icon size. An expressive visualization must allow the comparison of arbitrary icons. To guarantee this, icons must all be of uniform size so that icon features such as line lengths and area sizes are comparable. Therefore, it can be assumed that all icon contours have identical radii.

An apparent approach would be to choose the size small enough to be able to fit all icons into their respective areas without overlaps. However this means all icons will be as small as for the smallest area. It also means that the icons may be to small to be readable (cf. Figure 1), which would defeat the sense of placing them on a map. Automatically deriving a suitable icon size is difficult. In our prototype, the icon size can therefore be set interactively by the user.

Moreover, in the following we assume the map is given in vector format. The map areas are therefore given as polygons, with one information record per area.

ICON POSITIONING

Icon positioning is done in two stages:

- Preprocessing: The icons are first positioned on the map with respect to their reference points. These initial positions are found under consideration of cartographic design rules for map labels. Conflicts between icons are not considered at this stage.
- Positioning: All occurred conflicts are analyzed and processed in a specific order. The solution of conflicts is successively achieved in the neighborhood of individual icons by displacing them by a given threshold distance of r_{max}. The sum of all conflicts is taken as the value of an *objective function*. The cycle of conflict detection, sorting and processing is repeated iteratively until either a conflict-free solution or a minimum of the objective function was found, or other abort criteria hold.

Initial positions - Calculation of Reference Points

Finding good initial positions is not trivial. Map areas can be irregularly shaped, or even be comprised of spatially disjunct subareas. In these cases, simple means like using the center point of the area's bounding box will fail if that point lies outside of the area polygon. In [6] an overview of possible methods for reference point calculations is given. A good strategy, also used in automatic label placement, is choosing a reference point on a medial axis of the area polygon. A medial axis can be calculated by different means, e.g. from the Voronoi diagram of its vertices (the skeleton, Figure 2A). Points on this line are guaranteed to lie within the polygon and therefore are good candidates for an initial reference point. For each area, one or more candidate points are calculated. To choose the best candidate as initial reference point, candidates are evaluated.

In cartography, the evaluation for labels combines the position evaluation, i.e. how good the association to the labeled map feature is, and the occlusion evaluation measuring the amount of information loss by occlusion of other map features (symbols, border lines etc.).

Analogous to this, if more than one candidate for an reference point exist for a given icon, the candidate with the best evaluation is chosen as initial reference point. The evaluation primary considers the quality of the spatial reference visualization. It should express how certain the association of the icon to its area is. Therefore, the distance of the reference point to the medial axis' midpoint is used as the value for the position evaluation. For a discussion of possible alternate evaluation methods, see [6].

Since interactivity is a premise and the maps used in combination with icons are usually void of other features than area boundaries (cf. Figure 3), an occlusion evaluation with border lines is optional. It can increase the (cartographic) quality of the results, but at the cost of increased computational effort.

A special problem arises if map areas are comprised of several disjunct subareas. In these cases the initial reference point is chosen inside the largest of the subareas. The relation of the subareas must then be established by virtue of coloring or connecting lines.

Conflict Detection

When all icons have been placed in their initial reference points, the overlaps, or conflicts, between icons must be detected. The more pronounced the overlap, the more information from the affected icons are lost, i.e. that conflict is more sever. The severity of conflicts can be expressed by assigning a *conflict weight*.

If using bounding circles to approximate icon shapes, conflict detection means simply to check if the distance $d(p_i, p_j)$ between two reference points p_i, p_j is less than the double radius r_B of the bounding circles. As a conflict weight c_{ij}, the intersection area of the icon's bounding circles can be used.

The detected conflicts are solved by displacing icons. Therefore, the conflict situation in the local neighborhood of icons must be assessed. Besides *actual conflicts* arising from the overlap of icons, *potential conflicts* that could arise by moving reference points during the displacement step (cf. section 3.3) must be considered. The creation of new conflicts can be desired if the overall conflict situation is improved by it.

For this, a *conflict radius* around a reference point can be defined. The conflict radius is calculated from the bounding circle radius r_B and a given displacement threshold $r_{\max}$ and is $r_C = 2(r_{\max} + r_B)$. The displacement threshold $r_{\max}$ is used to constraint the neighborhood in which the conflict situation is locally improved (cf. section 3.4).

A data structure for this task is the *conflict graph*. In this graph, two nodes are connected by an edge if the corresponding reference points are less than the conflict radius apart. The edge weights correspond to the conflict weights c_{ij}. The conflict graph therefore contains information on which neighboring reference points must be taken into account when choosing an alternate position for an icon in order to find a local optimum.

Iterative Conflict Solution

During one iteration step conflicts can be solved by either of two strategies. One possibility is to define a fixed set of displacement directions. Here, the icons are processed sequentially. The processing order can be defined in two ways. One is to begin with icons in the largest areas first as these offer the most freedom for displacement. The second possibility is to process icons with the least total conflict first. The idea behind this is to first move icons on the perimeter of conflict concentrations away from the center of that concentration in order to gain more space for further improvement near the center.

For each icon, as many alternate positions as defined directions are calculated. An alternate position is obtained by moving the reference point in the particular direction by at most r_{max} without leaving the area polygon (Figure 2B). Each alternative is evaluated (cf. section 3.1). The best-rated alternate position is then selected as the new reference point. If none is better than the current position, no displacement takes place. After every displacement, the edge weights of the conflict graph are updated. By controlling overlapping in the conflict graph, it can be guaranteed the solution is always improved by a displacement operation. The advantage of this fixed directions approach is that alternate positions can be determined and evaluated easily.

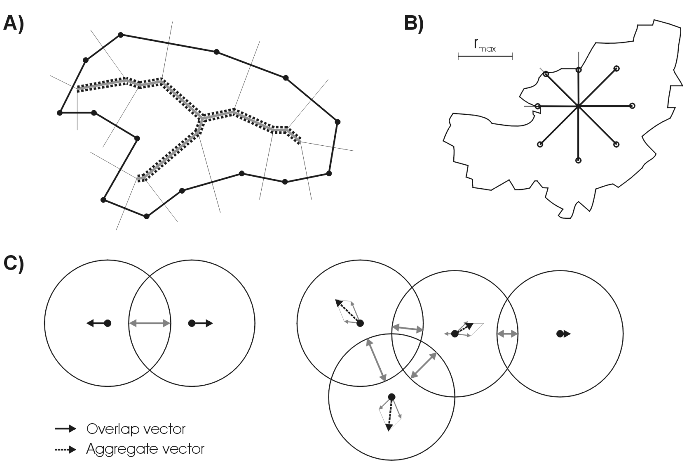
The second approach uses a model of repelling magnetic forces. If the bounding circles of two icon overlap, an *overlap vector* is constructed. Each reference point is then displaced in the direction of the overlap vector. The magnitude of these vectors is equal to half the distance required to completely solve that particular conflict (Figure 2C left).

If an icon is in conflict with more than one other, an overlap vector is constructed for each of these conflicts. The reference point is then displaced by the *aggregate vector*, which is constructed from the force parallelogram of the overlap vectors (Figure 2C right).

With this approach, both direction and distance of a displacement depend on the current distribution of reference points. A threshold r_{max} for the displacement vectors' magnitudes can be accounted for in a way similar to the global temperature constant in Simulated Annealing [5]. In each iteration, all reference points are displaced simultaneously, with both direction and distance depending on the current conflict situation. As a global method, displacements will not cause a degradation of the solution.

However, including a position or occlusion evaluation into this model requires the construction of additional repelling forces (constraints). The computational complexity is higher compared to the fixed directions approach where these evaluations are done independently from the calculation of the alternate positions.

Figure 2: (A) Choosing reference points: polygon skeleton (gray) with three medial axis (dashed), (B) fixed displacement directions and (C) magnetic force model



Iteration und Termination Criteria

An iteration step of the algorithm is complete when every reference point has been processed. If there are conflicts remaining, another iteration is executed.

The displacement distance threshold is reduced to $r'_{max} < r_{max}$. With increasing number of iterations, the reference points are displaced by ever decreasing distances. Therefore the algorithm starts with coarsely pre-positioning the icons, and then refines the solution locally in later iterations. The edge set of the conflict graph also has to be updated to reflect the new set of potential conflicts for r'_{max} (cf. section 3.2).

The algorithm terminates if one of three criterions holds:

- No conflicts remain, all icon overlaps have been resolved. The positioning problem has been completely solved.
- The value of the objective function did not change during the last iteration. A local minimum of the objective function was found as a solution.
- A specified number of iterations have been performed, or it is $r_{max} = 0$. By enforcing this criterion, an (approximately) constant runtime of the algorithm can be specified. This can be used to meet certain requirements regarding the overall system response time. The final distribution of icon positions is not necessarily a minimum of the objective function.

Implementation and Results

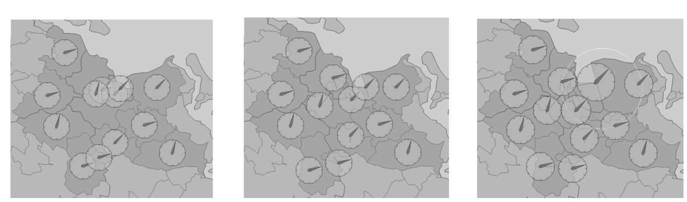
A prototype of the approach was integrated into an interactive information system for healthcare data, *LandVis$_t$* [13]. The user can choose from a variety of different icon techniques over a map of the administrative district boundaries of the German federal state of Mecklenburg-Vorpommern with no other objects recorded. For conflict solution, the algorithm works with eight predefined directions (cf. Figure 2B). The position evaluation is equal to the distance of the current reference point to the initial position. Figure 3 shows a map after initial positioning (left) and after optimization (20 iterations, center). The final positions yield some remaining icon overlaps. These are interactively solved by browsing the map with a lens (Figure 3 right).

SUMMARY

Map-based information systems in conjunction with icon techniques are well-suited for the graphical representation of spatial-dependant multivariate data. The icon positions must communicate the spatial reference of the data well, while at the same time be chosen in a way that minimizes information loss by icon overlaps. Most interactive systems do not properly regard these requirements, while cartographic methods strive for high-quality results that can not guarantee interactivity. In this paper an approach for interactive systems was presented that combines icon displacement with focus & context techniques to guarantee interactive response times.

As can be seen from Figure 3, the positioning is actually improved by the algorithm. However, as future work we plan to conduct studies to evaluate the results and to obtain statistical data on the degree of improvement. Moreover, we plan further extensions to the implemented system like the integration of a better occlusion evaluation to properly take into account the occlusion of other map objects (area boundaries, symbols).

Figure 3: Example of icon positioning and browsing with a cartographic lens



REFERENCES

[1] J. Ahn and H. Freeman. Autonap - an expert system for automatic map name placement. In *International Symposium on Spatial Data Handling*, Zürich, 1984.

[2] Jacques Bertin. *Graphische Semiologie : Diagramme, Netze, Karten*. de Gruyter Berlin, 1974.

[3] J. Bobrich. Ein neuer Ansatz zur kartographischen Verdrängung auf der Grundlage eines mechanischen Federmodells. *Deutsche Geodätische Kommission*, München, C455, 1996.

[4] A. C. Cook. Automated cartographic name placement using rule-based systems. PhD Thesis, Department of Mathematics and Computer Science, Polytechnic of Wales, Pontypridd, Mid Glamorgan, UK, September 1988.

[5] S. Edmondson, J. Christensen, J. Marks, and S. Shieber. A General Cartographic Labeling Algorithm. Manuscript, Harvard University, Cambridge, Massachusetts, 1996.

[6] G. Fuchs. Intelligente Ikonenpositionierung auf der Karte. Diploma Thesis, Fachbereich Informatik, Universität Rostock, 2002.

http://wwwicg.informatik.uni-rostock.de/Diplomarbeiten/2002/Georg_Fuchs/

[7] Havre, Hetzler, Nowell. ThemeRiver: Visualizing theme changes over time. In *Proceedings of InfoVis 2000*, IEEE Computer Society, pp. 115-123.

[8] S. A. Hirsch. Algorithms for Automatic Name Placement of Point Data. MSc Thesis, Department of Geography, State University of New York, Buffalo, 1980.

[9] E. Jäger. Untersuchungen zur kartographischen Symbolisierung und Verdrängung im Rasterdatenformat. PhD Thesis, Fachrichtung Vermessungswesen, Universität Hannover, 1990.

[10] Wolfgang Kresse. Plazierung von Schrift in Karten. PhD Thesis, Institut für Kartographie und Topographie, Friedrich-Wilhelms-Universität Bonn, 1994.

[11] W. Mackaness. An algorithm for conflict identification and feature displacement in automated map generalization. *Cartography and Geographic Information Science*, 4(21):219-232, 1994.

[12] Steven van Dijk. Genetic Algorithms for Map Labeling. PhD Thesis, Utrecht University, Department of Computer Science, November 2001.

[13] Christian Tominski. Visualisierungstechniken zur Analyse zeitlicher Verläufe über geographischen Karten. Diploma Thesis, Fachbereich Informatik, Universität Rostock, 2002.

http://wwwicg.informatik.uni-rostock.de/Diplomarbeiten/2002/Christian_Tominski/

[14] P. C. Wong and R. D. Bergeron. 30 Years of Multidimensional Multivariate Visualization. In G. M. Nielson, H. Hagen, and H. Müller, editors, *Sientific Visualization*, pages 3-33. IEEE Computer Society Press, Los Alamitos, 1997.

[15] S. Zoraster. Integer programming applied to the map label placement problem. *Cartographica*, 23(3):16-27, 1986.

Related Content